Dynamics of structural priming

Gaurav Malhotra

Doctor of Philosophy

The School of Philosophy Psychology
and Language Sciences
University of Edinburgh

2009

Abstract

This thesis is about how our syntactic choice changes with linguistic experience. Studies on syntactic priming show that our decisions are influenced by sentences that we have recently heard or recently spoken. They also show that not all sentences have an equal amount of influence; that repetition of verbs increases priming (the *lexical-boost effect*) and that some verbs are more susceptible to priming than others. This thesis explores *how* and *why* syntactic decisions change with time and what these observations tell us about the cognitive mechanism of speaking.

Specifically, we set out to develop a theoretical account of syntactic priming. Theoretical accounts require mathematical models and this thesis develops a sequence of mathematical models for understanding various aspects of syntactic priming. Cognitive processes are modelled as dynamical systems that can change their behaviour when they process information. We use these dynamical systems to investigate how each episode of language comprehension or production affects syntactic decisions. We also use these systems to investigate how long priming persists, how groups of consecutive sentences affect structural decisions, why repeating words leads to greater syntactic priming and what this tells us about how words, concepts and syntax are cognitively represented.

We obtain two kinds of results by simulating these mathematical models. The first kind of results reveal how syntactic priming evolves over time. We find that structural priming itself shows a gradual decay with time but the lexical enhancement of priming decays catastrophically – a result consistent with experimental observations. We also find that consecutive episodes of language processing add up nonlinearly in memory, which challenges the design of some existing psycholinguistic experiments. The second kind of results reveal how our syntax module might be connected to other cognitive modules. We find that the lexical enhancement of syntactic priming might be a consequence of how the modules of attention and working memory influence syntactic decisions.

These models suggest a mechanism of priming that is in contrast to a previous prediction-based account. This prediction-based account proposes that we actively predict what we hear and structural priming is due to error-correction whenever our predictions do not match the stimuli. In contrast, our account *embodies* syntactic priming in cognitive processes of attention, working memory and long-term memory. It asserts that our linguistic decisions are not based solely on abstract rules but also depend on the cognitive implementation of each module.

Our investigations also contribute a novel theoretical framework for studying syntactic priming. Previous studies analyse priming using error-correction or Hebbian learning algorithms. We introduce the formalism of dynamical systems. This formalism allows us to trace the effect of information processing through time. It explains how residual activation from a previous episode might play a role in structural decisions, thereby enriching our understanding of syntactic priming. Since these dynamical systems are also used to model neural processes, this theoretical framework brings our understanding of priming one step closer to its biological implementation, bridging the gap between neural processes and abstract thoughts.

Acknowledgements

A little over four years ago, Martin called me to his office and informed me that he had managed to secure the Dorothy-Hodgkin Postgraduate Award that would allow me to pursue my PhD. While it was Martin who finally secured the award, it was Johanna Moore who had initially introduced me to Martin and made an, ostensibly, convincing case for me. I am very thankful to her for this initial stimulus. The DHPA scholarship itself has been a very generous one. I owe my gratitude to someone, somewhere who must have taken pains to push this scheme through, allowing me to pursue these matters of largely academic interests.

Very little in the current thesis owes to what I did in the first two years. I attacked the problem in a very different manner than what is presented in this thesis and even though I was not getting many results, it took a lot of convincing from Holly, Martin and Jim for me to change my approach. The final blow was a conversation with Chris Williams who managed to persuade me that my project was too ambitious for the limited time afforded by a PhD. With hindsight, I think it was the correct decision to change my course – at least, with regards to finishing my thesis in time.

Throughout the course of my thesis (and much before) I have had many stimulating conversations with Keith Stenning. He has always made me ask the right questions and prevented me from losing perspective on what I am doing. I would like to thank him for always making time to see me and engaging with my research.

A lot in the final manuscript owes to my supervisor Martin Pickering. I would especially like to thank him for his patience – which allowed me to pursue the research in a calm and careful manner – and for his invaluable feedback on the drafts for the thesis. The thesis also benefited from discussions with and valuable comments from my other supervisors, Holly Branigan and James Bednar. I would also like to thank Naomi for helping me correct (a surprising amount of) grammatical errors and Janet for helping me with the stats.

Finally, I would like to thank my examiners Matt Crocker and Frank Keller who made the Viva a really pleasant experience and provided some important feedback on the thesis.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Gaurav Malhotra*)

To my parents,
for their untiring support.

The purpose of computing is insight, not numbers.
– Richard W. Hamming

Table of Contents

1 Introduction

1.1 Purpose and Character of the Investigation

This thesis lies on the cusp of two human activities. The first one is speaking, which allows people to exchange information with others. The second one is learning, which helps people accumulate information over time. Our intention is to examine how the act of speaking relies on learning.

On the surface, the two activities seem largely separate. The domain of one is external while the domain of the other is internal. Speaking is a social activity and involves two or more people while learning is largely a personal activity. However, beneath the surface, speaking is so intimately connected with learning that it seems impossible for the former to exist without the latter. After all humans are not born speaking a language, but must learn it through their environment over a period of time.

There is another way in which speaking relies on learning. When we participate in a conversation, we engage in a social activity which requires us to be informative to our listener and relevant to the conversation (Grice, 1975). When we are not informative or relevant, we break a social contract and the communication is unsuccessful. To be informative and relevant we must store information from the conversation in our memory and access this memory before we speak.

So the question is not *whether* speaking relies on learning – it is plain that it does – but *how* it relies on learning. What are the mechanics of this relationship? The difficulty in answering this question arises from the fact that both speaking and learning are complex processes. We will soon review evidence which shows that speaking is not a unitary process but a sequence of separate processes. Similarly, learning is the name given to a number of different forms of information storage. Learning can be from visual, auditory or contextual information; it can be transient or long-lived; it can

be the result of gradual accumulation over a series of events, or a large change in the system as a result of a single event. Understanding the relationship between speaking and learning involves deciphering a group of relationships between different forms of learning and stages of speaking.

Amongst the different stages of speaking, the one that will concern us most in this thesis is the stage of grammatical encoding. At this stage, speakers decide how they arrange words in an utterance. Not all arrangements are valid. *John gave the book to Mary* is acceptable, but *The book Mary John to gave* is not. Naturally, speakers must learn how to produce a grammatically valid arrangement of words. Usually, there will be many grammatically valid arrangements for an utterance. *John gave the book to Mary* is valid, but so is *John gave Mary the book*. How do speakers choose between these utterances? Do they choose on a whim, or do they choose based on what they have learnt previously about grammatical arrangements? Do their preferences change and if they do, what kind of information leads to this change? Some insight into these questions comes from experiments which investigate how speakers' choice of the grammatical form of an utterance is influenced by previous utterances that they have heard or produced.

Structural priming: Insight into how we learn

Let us be clear on the question we are discussing. We said we were interested in investigating the relationship between speaking and learning. We saw that this was a very broad question which led to a series of questions about the relationships between different forms of learning and stages of speaking. Therefore, we chose to focus on one stage of speaking – grammatical encoding – and one form of learning – that from previous utterances that the speaker has heard or produced. The first study to systematically investigate this relationship was Bock (1986), which observed how speakers' choice of grammatical form of an utterance varied based on the grammatical form of previous utterances during the experiment.

Bock (1986) found that subjects were more likely to choose a grammatical form such as *John gave Mary the book* over *John gave the book to Mary* when they had previously heard (and then produced) a sentence such as *The rock star sold an undercover agent some cocaine* as compared to when they had previously heard (and produced) a sentence such as *The rock star sold some cocaine to an undercover agent*. This finding shows a tendency in subjects to repeat the grammatical form, or syntactic structure, of

an utterance and has been called *structural priming*. A series of experiments conducted since this study have reproduced these results and repeatedly found that subjects tend to be structurally primed, not only after producing an utterance, but also after hearing it. Further studies also showed that structural priming increases when the verb repeats between the prime and target sentence – the *lexical-boost* effect (Pickering & Branigan, 1998) and that structural priming seems to last a long time (Bock & Griffin, 2000) while lexical boost decays quickly (Hartsuiker, Bernolet, Schoonbaert, Speybroeck, & Vanderelst, 2008).

Why do subjects show this tendency to repeat the syntactic structure of an utterance? Subjects seem to store information regarding the first utterance and use this stored information while producing the second utterance. But how exactly is this information stored? Is it stored in the same part of the cognitive system that is responsible for sequencing words into valid arrangements? Or is it stored separately as a memory of the whole utterance? How does this stored information evolve over time and why does it seem to last for a longer time as compared to its lexical boost? These are the core questions we will try to answer in this thesis. In answering these questions we will tease apart the processes and learning mechanisms that are involved in the act of speaking.

Method: Computational modelling

An analysis of mechanistic principles behind speaking can be made in two stages. The first stage is observation through psychological experiments and the second is interpretation of these results through systematic inference. One way of making systematic inferences is to develop a theoretical account which describes the computational processes that can generate the given data. We intend to develop such a theoretical account of structural priming. Theoretical accounts require mathematical models and this thesis develops a sequence of mathematical models. Each mathematical model tries to explain a set of experimental observations and outlines the assumptions required to replicate these observations.

In this thesis, we adopt the theory of dynamical systems to model the computational processes underlying language production. The theory of dynamical systems provides a mathematical apparatus to track the evolution of the system through time. Because we are interested in the temporal properties of structural priming – i.e. the effect of an episode of learning at different points in time – this theory helps us investigate the

influence of listening or speaking on future utterances. In the following chapters, we will see that a number of phenomena associated with dynamical systems (hysteresis, adaptation, competition) provide a formal way to think about the changes to our cognitive system as a result of linguistic processing. Dynamical systems have also been used to model neural processes underlying information processing in the brain, so adopting this framework brings the computational implementation of language production close to the underlying physiological processes.

Key assumptions and findings

In a series of four models, we explore how information can be stored as a result of linguistic processing and how this stored information decays. We treat linguistic processing (both comprehension and production) as a flow of information through the cognitive system. We claim that as information flows through the system, it gets transformed into different forms, suitable for representation in a set of disjoint modules. Each module is implemented as a dynamical system. As information flows through different modules, it stimulates the dynamical systems. Such stimulation could lead to a change in the state of these systems. In this manner, the dynamical systems record the effect of each flow of information during an episode of linguistic processing. We also assume that the behaviour of these dynamical systems depends on their existing state. Therefore, once the information during a particular episode is recorded as the state of a dynamical system, it interferes with subsequent episodes of linguistic processing. In this manner every episode of linguistic processing comes to depend on previous episodes.

Our first model shows that this scheme of recording information in a set of dynamical systems can lead to structural priming. We also explore how grammatical encoding during a particular episode of speaking depends on the choice of verb used in the utterance. We hypothesise that the relationship between the structure of an utterance and the verbs used in it could be modulated by the degree of *automaticity* in the system (i.e. how automatically we are making linguistic decisions while speaking). We show how varying this automaticity could lead to a change in the lexical enhancement of structural priming and could be a reason for the lexical boost effect.

Two subsequent models explore how dynamical systems that store information about linguistic processing could lose this information over time. We hypothesise that structural representations and their lexical context are recorded by two different kinds

of dynamical systems. We show that these two kinds of dynamical systems can have different longevities and this difference in longevity is responsible for a gradual decay in structural priming and a quick decay in its lexical boost. These models also look at two different kinds of learning mechanisms that could be at play during linguistic processing. One kind of mechanism records short-term information between a prime and a target trial, while the other mechanism accumulates learning over a series of trials. We show how a combination of these mechanisms can lead to the pattern of results observed in some psychological experiments.

Our final model extends the previous models and encodes the series of steps in which the different types of dynamical systems could be activated during comprehension and production. This model overcomes some of the computational limitations of the previous models and shows how symbolic structures can be represented in our mathematical models. By encoding the processes involved in comprehension and production, this model allows us to explore how comprehension and production might overlap and how the linguistic processing during one can structurally prime the other. This model also allows us to understand the role of a speaker's grammatical knowledge during comprehension and production. In our scheme, learning in modules does not change a speaker's grammatical knowledge and simulations of this model demonstrate that structural priming may be separable from a change in grammatical knowledge.

1.2 Organisation of the thesis

This thesis is divided into three parts. The first part consists of chapters 2 and 3 and provides the background to the thesis. The second part consists of chapters 4 and 5, which develop the mathematical models. Finally, chapter 6 provides a general discussion of the models and a review of the major findings.

We begin with *Chapter 2* which surveys the literature on language production with special emphasis on structural priming. We review the evidence for dividing language production into different stages. Of particular interest to us is the stage of grammatical encoding during which speakers arrange given concepts into a sequence. We will review the reasons for dividing this stage into a number of processes and show how experiments on structural priming provide a key investigative tool for studying grammatical encoding. The dual role of structural priming as both an investigative tool and a psychological phenomenon motivates our study of the learning mechanisms behind structural priming.

Ours is not the first attempt to develop a theoretical account of structural priming and in *Chapter 3* we review two popular accounts – one mathematical and the other conceptual. This chapter focuses on the types of learning mechanisms that can allow us to model structural priming and classifies each account based on these learning mechanisms. The first account uses error-based learning while the second account uses 'trailing-activation' to explain structural priming. Each account has its shortcomings – the first one fails to account for some experimental observations while the second lacks formal details. These shortcomings motivate the need for a formal account with an alternative learning mechanism.

In *Chapter 4* we start to develop a formal account of our own to explain structural priming and its properties. We present three models, in increasing order of complexity, that try to replicate several psychological experiments on structural priming. These models rely on the theory of dynamical systems and we begin the chapter by reviewing this theory and discussing how and why this theory can be used for modelling the processes of grammatical encoding. The discussion of each model is divided into three parts. The first part discusses the architecture of the model; the second part provides a formal description; the third lists the simulations performed on the model and the results. We discuss the results of all three models together at the end of the chapter.

Chapter 5 discusses several shortcomings of the models developed in Chapter 4. None of these shortcomings make the models incorrect, but they do leave these models open to criticism about their plausibility. Therefore, in Chapter 5, we develop an extended model which remains true to the principles of the models discussed in Chapter 4, but overcomes their limitations. This model sheds further light on the representation of information during language production and allows us to explore issues like the overlap between production and comprehension. The validity of this model is demonstrated by replicating several experimental findings related to structural priming.

Finally, *Chapter 6* sums up our findings and broadens the scope of our discussion from structural priming to the nature of learning and information flow in the human cognitive system. We discuss how the models presented in Chapters 4 and 5 contribute to the debate about the nature of learning during comprehension and production and the purpose of structural priming. We also identify the limitations of the models and the direction in which our work can be taken in the future.

The computational models reported in this thesis were developed using MATLAB™ and the code for the different models is available as a zip file on the attached CD-ROM.

2 Structural priming and language production

2.1 Introduction

The goal of this chapter is twofold: to outline the properties of structural priming and to review the importance of structural priming in the study of language production. The second goal provides the context and tells us why we study structural priming. The first goal then fills in the details and enumerates the gaps in our knowledge.

We begin by reviewing what we know about language production in humans and the place of syntactic decisions during production. There are a number of questions about making syntactic decisions that require careful consideration: at what stage are syntactic decisions made during language production; what information influences these decisions; how do we learn to make such decisions? We will see that there seems to be a consensus about how to answer some of these questions, while answers to others are as yet unknown, or controversial. This is where the study of structural priming comes in. Structural priming helps us understand the processes and representations that underlie syntactic decisions. Researchers have already conducted studies that show how syntactic decisions are influenced by other kinds of choices made during language production. We will be interested in two kinds of choices that can influence syntactic decisions – the choice of words used in an utterance and the choice of the meaning that the speaker wants to convey. It is not straightforward to answer how each of these choices can influence syntactic decisions. But we will see that studies of structural priming provide some insight into how syntactic decisions can be influenced by each of these choices.

Finally, investigations into the nature of structural priming also show how it changes with time. This property of structural priming provides crucial insight into the pro-

cesses responsible for its existence, the nature of learning syntactic choice and the interaction of syntactic representations with other kinds of information during language production. Therefore we will present a brief summary of the studies that show how structural priming changes with the passage of time. Together, these properties of structural priming and their role in syntactic decisions will motivate the need for a mechanistic account that can provide the etiology of structural priming in language production.

2.2 The Language Production System

Now let us look at the processes involved in language production. From the information processing perspective, production can be seen as a transformation that converts an input signal that encodes a mental state to an output signal that encodes speech. Somewhere along this transformation, speakers need to choose the information they want to convey, how to package this information and how to articulate it. Specifically, when speakers package information they need to make decisions about the syntax of their utterances. In this section, we will review what we know about the processes of making syntactic decisions. These processes can be viewed in two different contexts. The first context is that of the mental processes of the speaker – i.e. the set of processes in the speaker's brain that lead to a syntactic decision. The other context is that of a conversation between two or more people – i.e. the set of linguistic exchanges that lead to a syntactic decision. We will see that the study of syntactic priming is central to the understanding of syntactic decisions in both these contexts.

§ 2.2.1 The organization of language production

We have just stated that when we produce a speech signal we need to perform a transformation from a mental state to overt speech. This transformation consists of a number of functions that need to prepare different aspects of the speech signal. For example the cognitive system must prepare the concepts to be expressed, the words that express these concepts, the order in which these words are arranged, the function words that make this order unambiguous, the suffixes and affixes attached to each word, the sound of each word etc. However, just because all these functions need to be performed by a speaker does not mean that each of these functions corresponds to an independent process of the language production system. It could very well be that, for example, the

same process that chooses the content words in an utterance also chooses the function words. How can we be sure of the independent existence of these processes in the cognitive system and how can we understand how information passes between these processes?

One answer to this question comes from the study of spontaneously occurring speech errors. People occasionally make an error while speaking, where they say something that they did not intend to – e.g. *minx in spoonlight* in place of *sphinx in moonlight* (Fromkin, 1971). The mistakes made during these speech errors are not random but occur in particular patterns. These patterns have been carefully studied and tell us that the construction of our utterances obey a strong set of constraints (Meringer & Mayer, 1993; Fromkin, 1971; Garrett, 1975). Based, in part, on the study of such speech errors, the processes of language production have been divided into three major classes: *conceptualization*, *formulation* and *articulation* (Levelt, 1989). Different models differ in some of the specific details of these processes, but there seems to be a general agreement on the broad outline of these processes (Bock & Levelt, 1994; Levelt, 1989; Garrett, 1975, 1988).

Conceptualization consists of the processes that determine what the speaker wants to say. The result of this process is a *preverbal message* (Levelt, 1989) that is passed on to the second class of processes for formulation. This second class, formulation, consists of a set of processes that take the preverbal message and translate it into a linguistic form. This translation consists of three mutually independent sub-processes of *functional processing*, *positional processing* and *phonological encoding* (see Figure 2.2.1). Functional processing involves the selection of words and assignment of syntactic roles to these words. However, it does not involve the process of ordering the words themselves. This ordering of words is done at the level of positional processing. Finally, after the selected words have been ordered, their sounds are retrieved during a separate stage of processing known as *phonological encoding*. This finishes the process of formulation and results in a phonetic plan for the utterance which is then passed on to a class of processes responsible for articulation of this phonetic plan.

We noted above that the goal of this thesis is to look at the processes responsible for making syntactic decisions. These are the processes that occur at the levels of functional processing and positional processing. Together, they are known as the processes of grammatical encoding (Levelt, 1989; Bock & Levelt, 1994). In this section we will identify the details of what each of these processes needs to achieve in order to perform grammatical encoding. We will discuss two kinds of evidence for the identification of

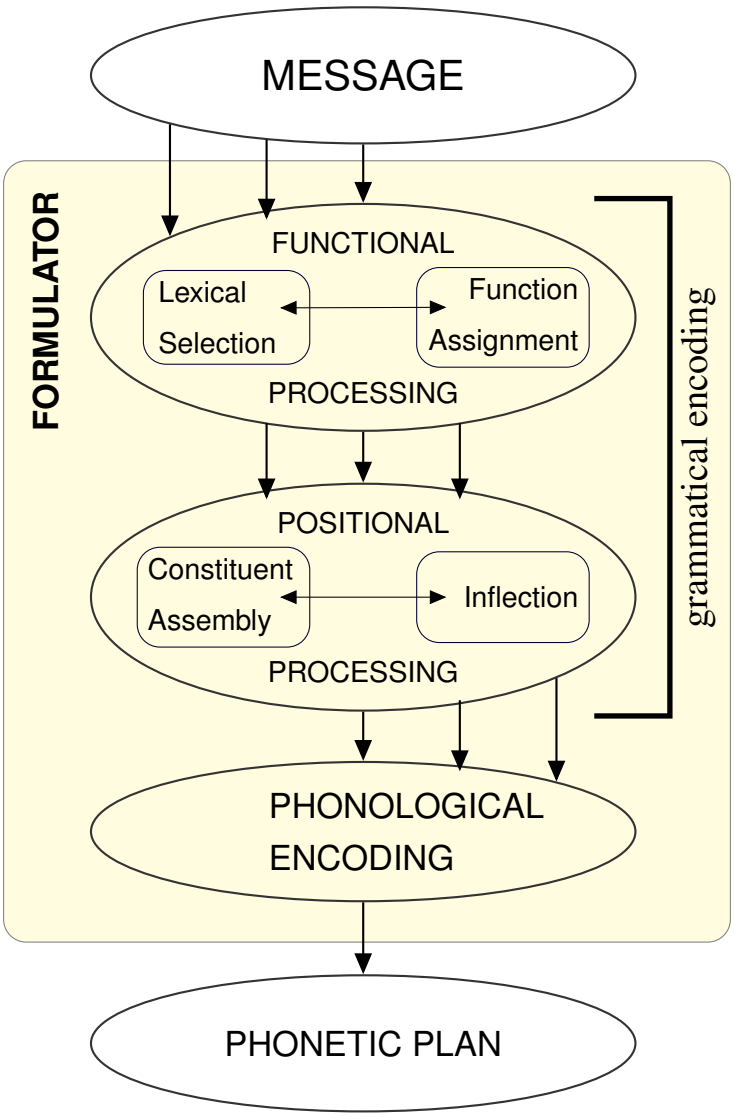


Figure 2.2.1: An overview of the language production system. Adapted from Bock and Levelt (1994) and Levelt (1989)

these processes. The first, as we noted above, is the evidence from speech errors. And the second, which is central to the theme of this thesis, is the evidence from the study of structural priming.

§ 2.2.1.1. **A word about structural priming.—** The phenomenon of structural priming is interesting for two reasons. Firstly, like speech errors, structural priming obeys certain constraints. The amount of structural priming varies based on the properties of the prime and the relationship between prime and target utterances. Thus structural priming allows the experimenter to determine those properties that play a role in determining the syntax of an utterance. Secondly, structural priming shows us

that the processes of language production are intimately tied to the processes of memorisation. It is memory that connects the target utterance to the prime. Since not every aspect of the prime utterance influences the syntactic structure of the target, structural priming helps us determine how the cognitive system extracts properties of utterances for memorisation. Thus, by analyzing the properties of structural priming, we can gain insight into the nature of the cognitive system underlying language production.

§ 2.2.2 Functional Processing

Functional processing comes immediately after conceptualization (see figure 2.2.1) and therefore gets a preverbal message as an input. It is also separated from positional processing, the stage that is explicitly responsible for ordering words. The goal of functional processing is to take this preverbal message and generate a list of words, each of which have been assigned its syntactic role. In order to do this, this stage relies on the speaker's *lexicon*. Therefore to understand the processes of functional processing we need to understand of the structure of the lexicon.

§ 2.2.2.1. **Lexical selection.—** Bock and Levelt (1994) identified two sub-processes involved at the stage of functional processing: *lexical selection* and *function assignment*. During lexical selection speakers retrieve those *lemmas* from the lexicon that allow them to convey the preverbal message. Lemmas are elements of the lexical network that lie between lexical concepts and word forms (Levelt, Roelofs, & Meyer, 1999) and contain grammatical information corresponding to each word. The meaning of words is represented at a separate level in the lexicon as lexical concepts. Evidence of this separation between the grammatical form and conceptual content of words in the lexicon is obtained through semantic substitution errors. When speakers make semantic substitution errors, they replace a word in an utterance with a semantically related word, as in *I received the trees, er — flowers you sent me*. The crucial fact about semantic substitutions is that they, nearly always, preserve the grammatical form class of the word that was replaced. If semantic substitutions are made by speakers when they make errors in retrieving the lexical concept, then this correct retrieval of the grammatical form class shows that speakers retrieve the grammatical form of utterances at a separate stage of processing. This second stage of processing that is responsible for retrieving the grammatical information corresponding to words is lexical selection. Thus semantic substitution errors show that lexical selection is part of grammatical encoding and independent of the processes that are responsible for retrieving the conceptual

content of a word.

That speakers access the semantic and grammatical properties of words separately from the phonological and morphological properties is further evidenced by the tip-of-the-tongue phenomenon. The tip-of-the-tongue phenomenon is a state in which a speaker cannot quite recall a familiar word but can recall words of similar form or meaning (Brown & McNeill, 1966). If speakers in the tip-of-the-tongue state can be shown to know certain grammatical property of the word, but be unaware of its phonological form, then clearly the two kinds of information must be separately accessed. This is exactly what was shown by Vigliocco, Antonini, and Garrett (1997) who found that Italian speakers can be aware of the syntactic gender of words for which they cannot yet generate a pronunciation code.

§ 2.2.2.2. **Function Assignment.—** Though lexical selection is able to give a list of words along with their grammatical class information, this list of words cannot yet be used to decide the order of the selected words. To do this, the speaker must determine the grammatical function that each word should perform in an utterance[1]. This task is performed by the process of function assignment. Bock and Levelt (1994) point out that the process of function assignment needs to be separate from lexical selection because the same words may serve different functions in different sentences – *The dog chased the cat* versus *The cat chased the dog*. They also point out that this process needs to be separate from positional processing because speakers can order words in two or more different ways while keeping the grammatical function of each word to be the same.

While we know that speakers need to assign syntactic functions independently of lexical selection and positional processing, we do not completely understand the mechanism of this functional assignment. Bock and Levelt (1994) identify four kinds of problems that must be addressed by a theory of functional assignment. They suggest that such a theory must identify (a) the nature of grammatical functions that are assigned, (b) the kinds of information that control the functional assignment, (c) the nature of the elements that the functions are assigned to and (d) the organisation of processes that carry out these operations. Each of these questions is the topic of much

[1]Following Bock and Levelt (1994), we use the term *grammatical function* to refer to the function of each lemma in an utterance. Examples of these functions are syntactic elements such as subject, verb, direct object and indirect object. But it is also possible to think of these syntactic elements as relations between different lemmas in an utterance. For example, in the utterance *Don Ciccio slapped on the kitchen table a wild rabbit*, *Don Ciccio* is 'the subject of the verb' *slapped*. In this sense, a subject is a *grammatical relation* rather than a grammatical function. In our discussion, we will assume that the two terms are equivalent and that the processes of functional assignment specify both kinds of relationships.

debate in linguistics and psycholinguistics (see Bock and Levelt (1994) for more details). In this thesis we will be concerned with the questions (b) and (d).

Let us first consider the question about the kinds of information that control functional assignment. At this stage of processing, the speaker has information available about the semantic and grammatical properties of the lemmas and about the message that needs to be expressed. This message contributes information about the semantic roles of different lexical concepts in an utterance. One way in which this information can be represented is as thematic roles such as `AGENT`, `PATIENT`, `THEME`, etc. Then, the question that can be asked is how this information about thematic roles along with the semantic and grammatical properties of the lemmas governs function assignment. A related question is whether this information governs just the function assignment, or if it directly governs the ordering of words in an utterance (discussed further below).

§ 2.2.2.3. **Investigation of function assignment using structural priming.—** Bock and Loebell (1990) used the structural priming paradigm to explore the role of conceptual information in function assignment and ordering of words. They gave subjects primes and targets that matched either in both the thematic roles and the phrase structure or just in the phrase structure. For example, the prime *The wealthy widow gave her Mercedes to the church* matches the target *The girl is handing the paintbrush to the boy* in both the phrase structure (noun phrase–verb–noun phrase–prepositional phrase) and the thematic role assignment (both have dative structure). On the other hand, this target would match another prime, such as *The wealthy widow drover her Mercedes to the church* in the phrase structure, but not in thematic role assignment (prime has locative structure). Bock and Loebell (1990) found that the additional overlap in thematic roles between prime and target had no enhanced effect on structural priming. Based on this result, they concluded that message level information such as thematic roles do not directly play a part in the ordering of utterances (i.e. the choice of phrase structure rules). This however leaves the possibility that message level information, such as thematic roles, do have an effect on functional assignment. This possibility was explored by another study, which also used the syntactic priming paradigm.

Bock, Loebell, and Morey (1992) varied the functional assignment of conceptual information in the prime and observed how it affected the target. Instead of varying the thematic roles like Bock and Loebell (1990), this study varied a primitive semantic feature – animacy – of subject and object arguments of the primes. A sentence such

as *The dog followed the car* has an animate subject and an inanimate object, while *The car followed the dog* reverses the grammatical function of the animate and inanimate entities. Bock et al. (1992) found an animacy priming – i.e. if subjects produced a prime with an animate entity as the subject, then they also tended to produce an animate entity as the subject in the target. Bock et al. (1992) also found that this animacy priming was separate and independent of the thematic role assignment – i.e. the amount of animacy priming received from a prime with an animate agent was equal to the amount of priming received from a prime with an animate patient. This finding led Bock et al. (1992) to argue that it is the primitive semantic features such as animacy which controls functional assignment, rather than message level information such as thematic roles.

But this observation still does not tell us whether these primitive semantic features influence just the functional assignment or if they directly influence the ordering of words in the utterance. To answer this question, Bock et al. (1992) observed that the amount of animacy priming is separate and independent, not just of the thematic role assignment but also of the phrase structure of the utterance. In other words, syntactic priming and animacy priming were additive and did not interfere with each other. This additivity of animacy and structural priming suggests that the influence of priming upon the choice of syntactic structure is independent of the influence of priming upon the choice of grammatical function (Pickering & Ferreira, 2008). Thus one can conclude that primitive semantic features, such as animacy influence function assignment and not the procedures that determine the phrase structure (i.e. ordering of words) of an utterance.

Beside the question about the kinds of information that control function assignment, the other important question was about the organisation of the processes that carry out function assignment. Bock and Levelt (1994) point out that functional assignment seems to be controlled by the verb. They suggest that, during function assignment, the lemma for the verb contributes argument structures which can be used to link the lemmas together. Studies on speech errors (Garrett, 1980) and attraction errors (Bock & Cutting, 1992) both suggest the centrality of the verb in performing function assignment. In chapter 5 we will discuss a *schema-based* representation for semantics that naturally explains this centrality of verbs in the assignment of grammatical functions.

§ 2.2.2.4. **Evidence for functional processing using speech errors.—** More evidence on the separation of functional and positional processing comes from the existence of *exchange errors*. During an exchange error a speaker switches two words in an utterance – *I received the flowers I sent you* instead of *I received the flowers you sent me*. Such an error shows that the speaker has mistakenly exchanged the syntactic functions for two lemmas, assigning them to incorrect roles in the utterance. Such exchange errors occur for both content words and function words. But importantly, content words almost always exchange only with other content words while function words exchange with function words (Harley, 2001) implying a separation between the processes that operate on the two types of words. Based on such observations, Garrett (1975) proposed that syntactic planning should be separated into functional and positional processing, with content words being assigned grammatical roles during the former and function words being chosen during the latter.

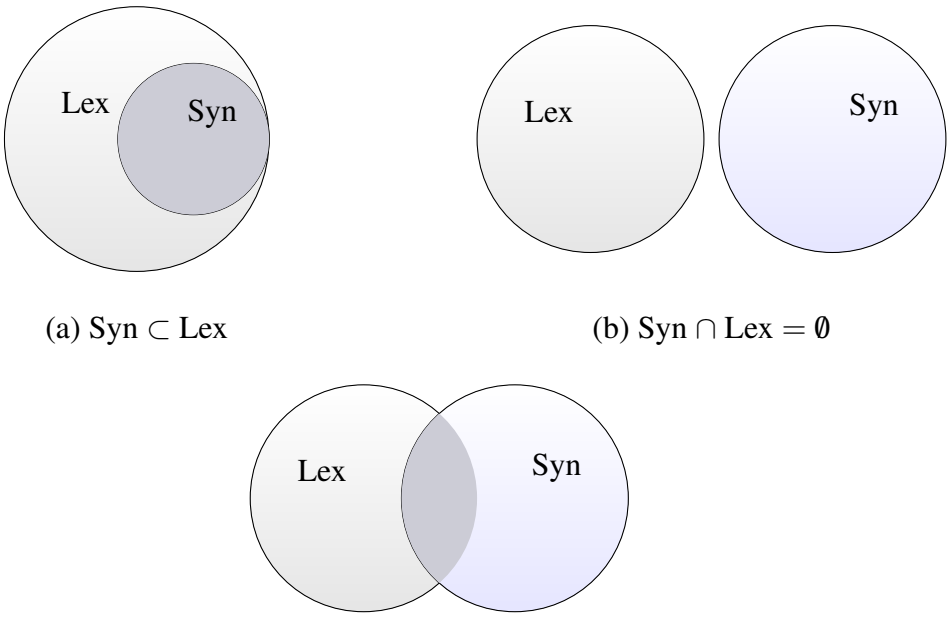
§ 2.2.3 Positional Processing

The procedures of grammatical encoding must not only associate each lemma with a grammatical function, they must also impose a sequence on these lemmas. This arrangement of lemmas in a sequence is done through positional processing, which consists of two kinds of operations: *constituent assembly* and *inflection* (see Figure 2.2.1).

§ 2.2.3.1. **Constituent Assembly.—** Let us first look at the operation of constituent assembly which is the actual process of ordering the lemmas into a sequence. For performing this operation, the speakers have, at their behest, the knowledge of the grammatical class of each lemma, the semantic features of the lemmas (obtained from the lexicon), the grammatical function of each lemma and the semantic properties of the message (e.g. the thematic roles). With all this information available, speakers use their *procedural knowledge* of the language to assemble the lemmas into a sequence. Just like the question about the role of different kinds of information in function assignment, we can also ask the role of different kinds of information in constituent assembly.

A related question is about the nature of the speaker's procedural knowledge – the knowledge of a list of procedures that allows the speaker to incrementally construct the utterance from the list of lemmas and under the constraints imposed by the speaker's language. A crucial property of this procedural knowledge is that it needs

to be lexically driven – i.e. the grammatical properties of the lemmas control the structure of the utterance and not the other way around. Levelt (1989, pp. 236–246) proposed a set of procedures based on a computational model developed by Kempen and Hoenkamp (1987) that can perform such a lexically-driven grammatical encoding. These set of procedures, called *categorical procedures*, incrementally construct hierarchical constituent structures based on the available lemmas and their grammatical categories. The critical question – a source of much debate in psycholinguistics – is that of the dependence of these categorical procedures, on the speaker's lexical knowledge. Three possibilities exist: the procedural knowledge is (a) completely dependent, (b) completely independent, or (c) somewhat dependent on the lexical knowledge. These possibilities are pictorially depicted as Venn-diagrams in Figure 2.2.2.



(a) Syn $\subset$ Lex

(b) Syn $\cap$ Lex $= \emptyset$

(c) Syn $\cap$ Lex $\neq \emptyset$

Figure 2.2.2: The sets 'Syn' and 'Lex' stand for the speaker's procedural and lexical knowledge. The symbols $\subset$ and $\cap$ refer to the subset and intersection operations and $\emptyset$ is the symbol for the empty set.

§ 2.2.3.2. **Investigation of constituent assembly using structural priming.—** Several studies show the independence of a speakers procedural knowledge. We saw above that Bock and Loebell (1990) showed the independence between procedural knowledge (the knowledge responsible for word ordering) and thematic roles and Bock

et al. (1992) showed the independence between procedural knowledge and primitive semantic features. Pickering and Ferreira (2008) discuss further evidence that suggests that something like phrase structure rules operate during sentence production and structural priming of these phrase structure rules is independent of variation in other aspects of the grammatical structure of sentences.

These studies seem to indicate the view in Figure 2.2.2 (b) where the procedural and lexical knowledge are completely independent. However, structural priming has also shown evidence of an overlap. Pickering and Branigan (1998) was the first study to look at the lexical influence on selection of syntactic structure. They used a structural priming paradigm which varied the lexical overlap in prime and target utterances and noted the effect on the syntactic form of the target. Like Bock (1986) and Bock and Loebell (1990), they found a structural priming effect so that subjects were more likely to use a syntactic structure for the target utterance, if the prime used the same structure. Crucially, they also found that this structural priming increased if the prime and target utterances used the same verb. This result, known as the *lexical-boost* effect, has been replicated by several other studies since (Branigan, Pickering, & Cleland, 2000; Corley & Scheepers, 2002; Cleland & Pickering, 2006; Schoonbaert, Hartsuiker, & Pickering, 2007). The increase in structural priming suggests that procedural knowledge is, at the very least, associated with lexical knowledge. Equivalently, one can argue that the process of constituent assembly seems to draw on the process of lexical selection.

Thus the overlap in lexical and procedural knowledge seems to be closest to Figure 2.2.2 (c), such that procedural information is partially abstract and partially associated with lexical entries. Pickering and Ferreira (2008) suggest that this partial overlap between lexical and syntactic information can be represented by two types of accounts: (a) A two-locus account which assumes that separate cognitive systems lead to abstract (lexically independent) priming and lexicalized (lexically boosted) priming, and (b) a one-locus account which assumes a single mechanism can explain both abstract priming and the lexical boost. In the next chapter we will present, in detail, two accounts that claim to explain structural priming: an error-based account which is a formal account showing how the lexically independent component of priming can arise during language learning and a trailing activation account which is a conceptual account that proposes a mechanism for lexically boosted priming.

So far we have considered priming studies that investigate influence of lexical and semantic information on constituent assembly. The first set of studies demonstrated that the process of constituent assembly is uninfluenced by thematic roles and primi-

tive semantic features like animacy. The second set of studies showed that constituent assembly is influenced by lexical information. Cleland and Pickering (2003) extended these findings by observing that structural priming is enhanced by not only the exact repetition of the lemma between prime and target trials (as observed by Pickering and Branigan (1998)), but also by the repetition of a semantically related lemma. Thus structural priming shows not only a lexical boost, but also a "semantic boost". This finding implies that the process of constituent assembly can be influenced by the semantic features of the lemmas in the utterance and therefore, on the surface, seems to contradict the conclusions of Bock et al. (1992), who found that primitive semantic features do not affect choice of syntactic structure. However, Bock et al. (1992) were interested solely in those semantic features that affect the mental prominence, or *conceptual accessibility* of words. They found that this conceptual accessibility influences choice of grammatical function but does not influence the choice of phrase structure. The results of Cleland and Pickering (2003), on the other hand, used category co-membership (*dog-cat*) to measure semantic relatedness and found that lemmas related in this manner do show an influence on the choice of phrase structure. In addition, Cleland and Pickering (2003) also found that phonological relatedness (*cat-cot*) does not influence structural priming, suggesting that the process of constituent assembly is unaffected by phonological feedback.

While the results of Cleland and Pickering (2003) do not contradict the findings of Bock and Loebell (1990) and Bock et al. (1992), Pickering and Ferreira (2008) note that two other priming studies challenge the belief that message structure does not influence the choice of syntactic structure. The first study done by Griffin and Weinstein-Tull (2003) manipulated the message structure of primes. They found that manipulating the number of thematic roles in the prime had an effect on whether or not subjects paraphrased a finite complement clause such as *John believed that Mary was nice* as a noun phrase plus an infinitive clause *John believed Mary to be nice*. Griffin and Weinstein-Tull (2003) concluded that, subtle differences in the message structure can affect how speakers grammatically encode message elements. The second study was conducted by Chang, Bock, and Goldberg (2003) and also varied the message structure of the prime, while keeping the phrase structure to be the same. For example, the two primes *The man sprayed the water on the wall* and *The man sprayed the wall with water* have the same phrase structure: noun phrase–verb–noun phrase–prepositional phrase, but differ in their message structure. The first phrase adopts a *theme-locative* structure, putting the theme (*water*) before the locative (*wall*), while the second adopts

a *locative-theme* structure putting the locative before the theme. Chang et al. (2003) found a difference in priming patterns for the two kinds of primes, with theme-locative increasing in likelihood after a theme-locative prime and the locative-theme structure increasing in likelihood after a locative-theme prime. Because there is no difference in the phrase structure of the two types of primes, the difference in structural priming must be due to the difference in the message structure. Thus, in contrast to Bock and Loebell (1990), these results show that thematic roles can influence the mechanism of constituent assembly, under the condition that both primes show a similar amount of structural priming.

§ 2.2.3.3. **Inflection.—** The processes of inflection are responsible for encoding grammatical features of lemmas such as tense, aspect and number and determining how words are bound to other words in the utterance. Evidence for the independent existence of this subsystem comes from speech errors. In an exchange error like *She's already **trunk**ed two **pack**s* (Garrett, 1975), the content words *trunk* and *pack* get exchanged, but their bound-suffixes do not. This phenomenon, known as *morpheme stranding*, is known to occur in spontaneous speech (Garrett, 1975) and shows that the ordering of content words and the attachment of their suffixes (or, more generally, the process of inflection) occur at different stages in language production. Besides speech errors, there is also evidence from affix loss in Broca's aphasia and affix addition to neologisms in jargon aphasia that suggests an independent subsystem in the brain responsible for inflection (Harley, 2001).

§ 2.2.4 Production and dialogue

So far we have been viewing language production as an information processing system that transforms a preverbal message into overt speech. But language production is rarely done in isolation. During a dialogue, two such language production systems interact and the processing of one system starts to depend on the other. Speakers do not just want to produce an utterance, they want to produce an utterance for a listener. This means that speakers must tailor their utterances so that they are relevant to the conversation and understandable for the listener. Thus, the speaker's production system comes under an additional load trying to respond to the listener in a limited time frame and keeping in mind the listener's point of view. Garrod and Pickering (2004) note that due to these factors and others (such as the use of fragmentary and elliptical utterances in dialogue), dialogue should be much more difficult than the simple act

of production, or monologue. However, they note, that this does not seem to be the case. Speakers find dialogue just as easy, if not easier, than monologue. In light of this observation, what can we say about the processes of language production during dialogue?

One answer comes from the study of dialogue as a joint activity between interlocutors (Clark, 1996). According to this theory, interlocutors divide the burden of holding a conversation amongst themselves. Instead of interlocutors behaving in the same way as they would have during a monologue, they reduce their work by depending on their partner for generating part of the information required for holding the conversation. Specifically, Clark (1996) proposed that interlocutors maintain a "common ground" which includes the shared information between them and their partners[2]. This common ground could contain, for example, knowledge of the social and linguistic community that they both form part of and new information that interlocutors have exchanged during the conversation. By consulting the common ground, speakers can then decrease their effort during a dialogue by only providing novel information that adds to this common ground and by allowing their partner to consult the common ground to generate the complete information structure. In this way, the language production system of the speaker can defer part of its load to the listener, allowing them to consult the common ground to fully understand the speaker. However, Garrod and Pickering (2004) point out that actively maintaining this knowledge of the common ground in addition to the speaker's private knowledge would in itself create an additional burden for the speaker and therefore does not completely tell us how speakers are able to participate in dialogues with such ease. In addition, there are other activities in a dialogue, such as listening to one's partner at the same time as planning one's speech, that should make dialogues harder and cannot be explained by appealing to the notion of common ground.

Some crucial insights into 'why dialogue can be easy' come from studies of repetition in dialogue. Levelt and Kelter (1982) found that during a conversation, interlocutors tended to repeat material used by their partner. Speakers answered questions such as *(At) what time do you close* with *Five o'clock* or *At five o'clock* depending on whether the original question used *At* or not. This repetition could be due to repetition of a function word, or due to structural priming (of prepositional or non-prepositional structure). Garrod and Anderson (1987) found repetition at another level. They made participants describe locations in a cooperative maze game. They observed that when a

[2]Specifically, information that interlocutors *realise* is shared between them.

player used a description like *I'm two along, four up* for intimating their location on a maze, their partner tended to use a similar 'path-description' scheme – *I'm one along, five up*. On the other hand, when a participant used a coordinate scheme of describing their location – *I'm at B4* – their partner also tended to use the same scheme – *I'm at A5*. A third kind of repetition was observed by Brennan and Clark (1996) who asked subjects to describe pictures to each other and found that interlocutors tended to repeat the referring expressions used by their partner.

One reason behind this repetition could be a strategic choice by the interlocutor to repeat the words, syntax or description scheme of their interlocutor. But another reason could be an automatic priming mechanism. Branigan et al. (2000) put this second hypothesis to test by checking if interlocutors showed structural priming based on the choice of syntactic structure made by their partner in a dialogue. They used a confederate-scripting scheme where one speaker was a confederate of the experimenter and produced scripted descriptions that systematically varied the syntactic structure. Participants were required to take turns to describe pictures to each other. They found that participants in this dialogue experiment tended to choose a syntactic structure if they had just heard their partner (the confederate) use that syntactic structure. For example, if a participant had just heard their partner describe a picture as *The burglar gave the priest a camera* (double-object phrase), they tended to describe their picture as *The postman offered the policeman a banana* rather than *The postman offered a banana to the policeman* (prepositional-object phrase). In other words, participants in a dialogue showed structural priming. Branigan et al. (2000) also found that participants showed lexical boost – the amount of structural priming increased significantly when the picture described by the confederate and the subsequent picture described by the participant used the same verb. Other studies have subsequently replicated this effect of structural priming in dialogue (see Pickering and Ferreira (2008) for a review).

These findings of repetition and priming in dialogue illuminate another reason which might be responsible for why speakers find dialogue to be easy. Because interlocutors show a tendency to repeat the linguistic structure they have just heard, they can avoid constructing each of these linguistic structures from the ground-up and reuse the linguistic structure constructed by their partner. Based on these observations, Pickering and Garrod (2004) have proposed an *interactive-alignment* account of dialogue. This account suggests that interlocutors maintain an "implicit common ground" instead of separately maintaining their internal knowledge and a common ground for the conversation. While common ground depends on both the shared information and

on the knowledge that this information is shared, implicit common ground depends simply of the shared information. Pickering and Garrod (2004) suggest that interlocutors can generate this implicit common ground by aligning different levels of of their linguistic representations with those of their interlocutor. This alignment of linguistic representations can be achieved automatically through the mechanism of priming. In this interactive-alignment account, interlocutors can use their own (*aligned*) representations to produce their utterances, without explicitly maintaining the state of their listener's knowledge all the time. In this sense, the interactive-alignment account proposes an implicit common ground generated through processes of priming.

Another property of structural priming plays a crucial role in the interactive-alignment account. As noted above, Branigan et al. (2000) found a lexical boost in structural priming during dialogue. The interactive-alignment account proposes that such a lexical boost is one manifestation of a more general phenomenon whereby alignment at one level spreads to alignment at another level. Just like lexical alignment can spread to syntactic representations by boosting certain syntactic choices over others, Pickering and Garrod (2004) argue that alignment at other levels can spread in a similar fashion. This mechanism of spreading alignment ensures that alignment is not just a localised phenomenon and that alignment of certain linguistic representations eventually leads to alignment of mental states. In agreement with this hypothesis, Cleland and Pickering (2003) have found that semantically related nouns between prime and target also led to an increase in structural priming, showing that lexical boost is not an isolated instance of alignment spreading from one linguistic level to the other.

Criticism of the interactive-alignment account comes from studies that demonstrate that interlocutors can use strategic processes to tailor their utterances to their partners in dialogue. Brennan and Clark (1996) asked subjects to participate in a card-description task and showed that interlocutors aligned their lexical descriptions with that of their parter – a phenomenon called *lexical entrainment* (Garrod & Anderson, 1987). Brennan and Clark (1996) found that lexical entrainment was partner-specific – i.e. the effect of lexical entrainment became weaker when the partner of the speaker in a conversation was changed. Based on these partner-specific effects in lexical entrainment Brennan and Clark (1996) argued that interlocutors make implicit "conceptual pacts" between themselves in a dialogue (agreeing to use a particular term for the description of a particular card, in this case). They added that parter-specificity in their experiments showed that interlocutors explicitly referred to these conceptual pacts while constructing their descriptions. On basis of these arguments, Brennan and Clark (1996)

suggested that interlocutors rely on strategic, and not automatic, processes during language production (but see Garrod and Pickering (in press) for an objection to their experiment design). An interesting study in this context is that done by Horton and Keysar (1996) who observed that speakers took their partner's perspective into account when they were under no time pressure. But when the speakers were put under a time-pressure, they tended towards more egocentric descriptions without considering their partner's perspective. In chapter 4 we will discuss a model that offers a solution to these contrasting findings and is able to show a variable amount of priming based upon the level of automaticity in the system.

While the processes of language production during dialogue remains an active topic of debate, our discussion shows how priming, in general, and structural priming, in particular, is pivotal to the investigation of these processes. Priming forms the crucial causal link that connects comprehension to production, allowing production to reuse information from comprehension and, in turn, allowing the speaker to use information generated by their partner. In this way priming allows interlocutors to pursue dialogue as a joint activity.

2.3 Temporal properties of structural priming

So far we have considered structural priming as an investigative tool. We can control this tool by systematically varying the prime and observing the change in the target. The prime and target are separated in time and therefore priming is a mechanism that allows information to be transferred through time. One can think of priming as an information channel that stretches through time. The experimenter systematically varies information at one end of the channel (the prime) and observes the effect on the other end (the target). Different linguistic properties of the prime are transferred differently across the channel and therefore tell us about the nature of the channel itself. Take lexical boost as an example. When the prime and target contain the same word, we observe an increase in the flow of information along the channel. This increase in the flow of information tells us that words carry part of structural information, predicting a lexical influence on structural representations.

In this thesis, we would like to investigate the properties of structural priming itself. In the information-channel metaphor, we would like to investigate the mechanism of information-flow across this channel. One way to understand the properties of the channel is to increase or decrease its length and observe how it changes the flow of

information. Because the channel exists across time (between prime and target), increasing or decreasing its length is the same as varying the duration (or intervening utterances) between the prime and target and observing how it affects structural priming. The information channel that we have been talking about so far is nothing but a kind of memory. We want to investigate the mechanistic principles behind this memory and we are suggesting that one way to do this is to vary the time between encoding and recall and see how it affects memory.

The temporal properties of structural priming are of interest for another reason. There is an ongoing debate about the cognitive function of structural priming (Ferreira & Bock, 2006; Pickering & Ferreira, 2008). We saw above that one function of priming could be to facilitate alignment in dialogue. This account claims priming leads to aligned representation at a particular linguistic level and this alignment spreads to other levels leading to aligned mental states between interlocutors (Pickering & Garrod, 2004). A contrasting cognitive function of structural priming could be to perform language learning. According to this account, comprehension leads to implicit learning in the procedural knowledge and structural priming is a consequence of this implicit learning. Therefore, this account proposes that the function of structural priming is language acquisition. Chang et al. (2006) have developed an error-based model that tries to acquire such procedural knowledge and is able to show structural priming. (We discuss this model in greater detail in the next chapter.) Thus according to the alignment account, structural priming is the result of a memory trace that is required during the conversation, while according to the implicit learning account structural priming is the result of implicit learning that leads to language acquisition over a long period of time. We will revisit this debate from time to time during this thesis and flesh out the details and assumptions of each argument. For the present purposes, it is important to note that the temporal properties of priming are at the heart of this debate. In this section we review three temporal properties of priming and the experimental evidence supporting or opposing each. We will review two of these experiments in some detail because the models presented in this thesis are tested under the setup designed for these experiments.

§ 2.3.0.1. **Decay of structural priming.—** There is some recent evidence which shows that structural priming is long-lived. Bock and Griffin (2000) tested the duration of structural priming by varying the number of unrelated filler utterances that separated prime and target trials. In their first experiment subjects received 0, 1 or

2 filler utterances between prime and target trials. Bock and Griffin (2000) found no reduction in priming when the number of fillers increased. In their second experiment, they extended this finding by increasing the number of fillers between prime and target. Subjects saw 0, 4 or 10 intransitive (filler) utterances between the prime and target trial. They found no consistent decline in the amount of structural priming even with 10 filler trials between prime and target. Bock and Griffin (2000) argued that these findings supported the explanation of structural priming in terms of long-term learning rather than a transient memory activation mechanism. Bock, Dell, Chang, and Onishi (2007) replicated these findings using only auditorily presented primes and argued for the persistence of structural priming regardless of the modality in which the language structure is experienced.

However, in contrast to these findings Branigan, Pickering, and Cleland (1999) found a steady reduction in priming when primes were separated by 0, 1 or 2 fillers. Two aspects of this study differed from the experiments conducted by Bock and Griffin (2000). Firstly, participants in this study wrote down, rather than spoke, the prime and target sentences. Secondly, in the study conducted by Bock and Griffin (2000) primes and targets used different verbs while in the study conducted by Branigan et al. (1999) primes and targets used the same verbs. As we will see below, this overlap in the verb between prime and target proves to be a crucial factor in how priming decays.

§ 2.3.0.2. **Relative decay of structural priming and lexical boost.—** In order to understand the conflicting findings of Bock and Griffin (2000), who found structural priming to be long-lived, and Branigan et al. (1999), who found a quick decay in structural priming, Hartsuiker et al. (2008) conducted a set of experiment to see (a) if structural priming and lexical boost decay at different rates and (b) if the decay in structural priming depended on the modality (written or spoken) in which the experiment was conducted. Each experiment was divided into two sessions – one for testing written and the other for testing spoken conditions. One of the sessions of the experiment was conducted using the confederate-scripting paradigm similar to the one used by Branigan et al. (2000). This session tested the spoken modality and two subjects (one of whom was a confederate of the experimenter) took turns to describe pictures to each other. In order to test the results for the written modality, a second session was conducted using computer-based chatting. The procedure of the experiment remained the same – participants took turns to describe pictures to each other – except this time the descriptions were typed into the chatting program and one of the participants (the

confederate) was simply simulated using a computer script.

In both the sessions, Hartsuiker et al. (2008) varied the number of filler trials between the prime and target trials. Specifically, they inserted 0, 2 or 6 filler trials, called Lag 0, Lag 2 or Lag 6 condition. The pictures used for the prime and target trials could be described using either a prepositional-object dative (*The teacher sells an apple to the monk*) or a double-object dative (*The teacher sells the monk an apple*), while the pictures presented during the filler trials displayed actions that can be described using a transitive sentence (*The swimmer chases the police office*). In addition, Hartsuiker et al. (2008) also varied the lexical overlap between prime and target trials. Prime and target trials could either show pictures with the same action giving the 'Same Verb' condition, or they could show pictures with different actions giving the 'Different Verb' condition.

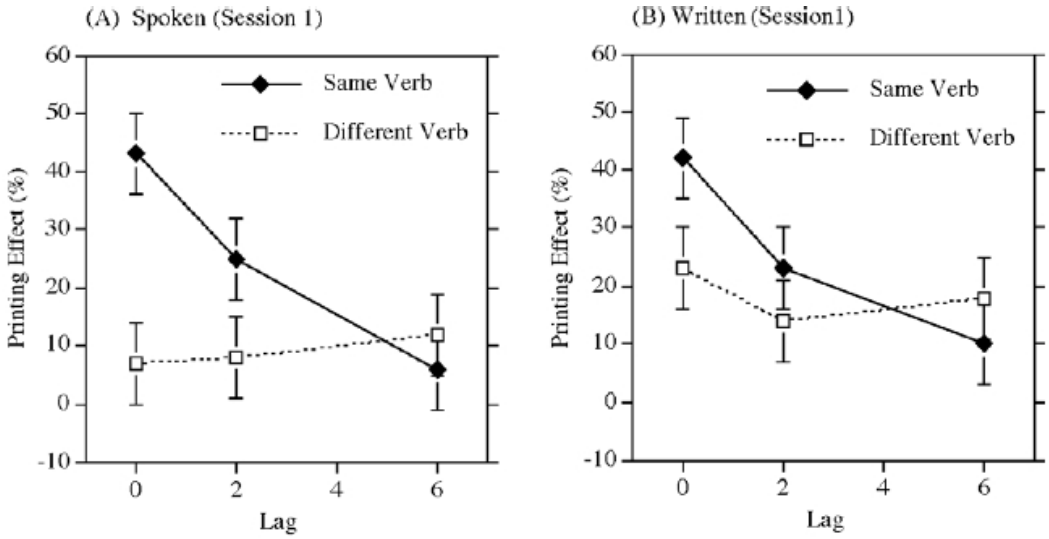


Figure 2.3.1: Results for a spoken and written session from Hartsuiker et al. (2008)

Hartsuiker et al. (2008) found, for both spoken and written conditions, structural priming persisted across all three lags and there was no numerically consistent trend in the change of structural priming. They also found that the 'Same Verb' condition showed larger structural priming as compared to the 'Different Verb' condition at Lag 0 – i.e. participants showed a lexical boost. However, in contrast to structural priming, this lexical boost was short-lived, with no reliable difference between the 'Same Verb' and 'Different Verb' conditions at Lag 2 and Lag 6. This result remained true irrespective of whether the participants used spoken or written modality. Based on these results, Hartsuiker et al. (2008) argued that the discrepancy in the longevity of structural priming observed by Bock and Griffin (2000) and Branigan et al. (1999) was

not due to the different modalities tested by the two experiments, but due to the fact that Bock and Griffin (2000) tested only Different Verb conditions while Branigan et al. (1999) tested only Same Verb conditions. A sample of these results for both spoken and written modalities are shown in Figure 2.3.1.

§ 2.3.0.3. **Relative accumulation of structural priming and lexical boost.—** The experimental evidence presented so far has measured the longevity of priming in terms of the fillers between prime and target trials. Each of these fillers used a different syntactic structure to that used during the prime trials. For example, the prime trials in Hartsuiker et al. (2008) could be described using prepositional-object or double-object datives, while the filler trials were described using a transitive sentence. Thus, the structural representations evoked during the filler trials do not interfere with those encoded during the prime trial. Thus these experiments can be seen as measuring a non-interference based decay in memory. A similar point is that these experiments measured the long-term effect of structural priming (and lexical boost) obtained from a single prime trial and not one accumulated over a series of trials.

Kaschak and Borreggine (2008) conducted experiments that also measure the long-term effect of structural priming and lexical boost, but instead of measuring the long-term learning obtained from a single trial, their experiment measured the effect of a series of trials. Each participant in the experiment received a list of trials, each of which consisted of an incomplete sentence. Participants were asked to complete these sentences. This sentence-completion task is in contrast with the confederate-scripting task used by Hartsuiker et al. (2008). Whereas the confederate-scripting task emulates a real-world dialogue situation, this task emulates a monologue situation. However, the results of priming have been shown for both monologue and dialogue and sentence-completion is the same task that was used by Pickering and Branigan (1998) in their demonstration of structural priming and lexical boost.

The experiment trials were divided into two phases, a *Training phase* and a *Testing phase*[3]. During the training phase, participants received ten prime stems such as:

- Megan gave her mom... (double-object prime)
- Megan gave a kiss... (prepositional-object prime)

Thus each prime stem was either a prepositional-object (PO) phrase or a double-object

[3]The terms *Training phase* and *Testing phase* are our own. Kaschak and Borreggine (2008) call these phases the *Bias phase* and *Priming phase*, respectively. But their terminology is confusing as primes are presented during both the phases.

(DO) phrase. This sequence of prime stems was used by Kaschak and Borreggine (2008) to cause long-term structural priming. Participants either received prime stems that elicited an equal number of PO and DO completions (Equal condition) or elicited completions of only one kind – either all PO or all DO (Unequal condition).

Kaschak and Borreggine (2008) were interested in measuring how this long-term memory created in the training phase would affect structural priming. In order to do this, they gave participants a list of six prime-target pairs at the end of the training phase. These sequence of six prime-target pairs were termed as the *Testing phase*. Each prime and target trial again consisted of a sentence-completion task. All prime stems elicited the same structure (either PO or DO), while the target stem could be completed using either of the two structures. Kaschak and Borreggine (2008) arranged their materials in such a manner that the participants who were biased towards one structure during the training phase received the opposite kind of prime during the testing phase. Thus a participant could be either in the Equal or Unequal conditions during the training phase and receive PO or DO primes during the testing condition. Here are examples of each of the resulting four (2×2) conditions:

Equal-PO [DO–PO–DO–PO–DO–DO–PO–PO–PO–DO] [PO–target . . . PO–target]

Equal-DO [DO–PO–PO–DO–DO–PO–PO–DO–DO–PO] [DO–target . . . DO–target]

Unequal-PO [DO–DO–DO–DO–DO–DO–DO–DO–DO–DO] [PO–target . . . PO–target]

Unequal-DO [PO–PO–PO–PO–PO–PO–PO–PO–PO–PO] [DO–target . . . DO–target]

where each prime stem is shown by the type of structure it elicits. We have enclosed the training and testing phrases within square brackets just for illustrating where the boundary lies.

In addition, to measure the relative longevities of priming and lexical boost, Kaschak and Borreggine (2008) divided their participant under two further condition – the 'Same Verb' and 'Different Verb' conditions. Participants under the Same Verb condition were given verbs picked from the same set during training and testing phases, while participant under the Different Verb condition received verbs from different sets in the two phases. Kaschak and Borreggine (2008) recorded the completions for the target sentences and calculated the relative proportion of PO and DO completions under each of these conditions. Their results are shown in figure 2.3.2.

Kaschak and Borreggine (2008) found that subjects under the Unequal condition showed lesser structural priming as compared to the Equal condition. Since the unequal

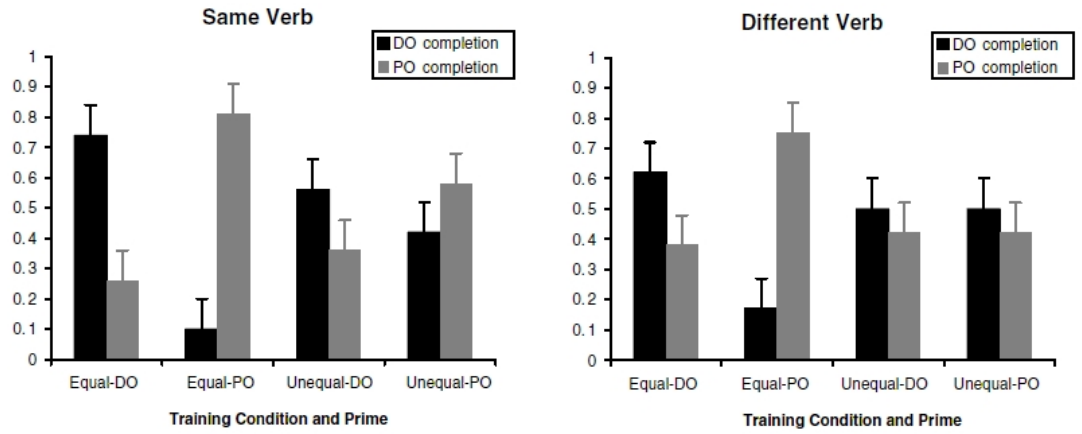


Figure 2.3.2: Results for priming under Same and Different Verb conditions from Kaschak and Borreggine (2008). The x-axis shows different (long-term) priming conditions and the y-axis shows the proportion of each type of completion.

condition was used to bias the subjects towards a structure that was opposite of the one they received during the testing phase, Kaschak and Borreggine (2008) argued that this was evidence for a long-term priming from the sequence of primes a subject completed during the training phase. Furthermore, they also found that this difference in priming was comparable for the Same Verb and Different Verb conditions. While the relative frequencies of PO and DO constructions during the training phase affected the magnitude of structural priming, Kaschak and Borreggine (2008) argued that this pattern was relatively unaffected by whether the verbs in training and testing phases were same or not, leading them to conclude that long-term structural priming is not greatly affected by the patterns of experience with particular verbs.

It is worth reiterating that this 'long-term' priming measured by Kaschak and Borreggine (2008) is quite different in nature from the long-term priming measured by Hartsuiker et al. (2008), or indeed Branigan et al. (1999) and Bock and Griffin (2000). While those experiments measured the decay in structural priming obtained from one trial, this experiment measured priming obtained from a series of trials (in the training phase). These two types of priming might internally rely on two different mechanisms of memory. We will explore this question in greater detail during the design of formal models for structural priming.

In any case, the different studies discussed in this section all agree that structural priming seems to be a relatively long-term phenomenon and seems to last for the duration of at least six to ten lags. Lexical boost, on the other hand, seems to decay relatively quickly. These findings suggest that structural priming and lexical boost rely on

somewhat different, even if overlapping, mechanisms of learning. In the information-channel parlance discussed at the beginning of this section, structural priming and its lexical enhancement seem to rely on different sub-channels of information transfer. This observation forms a key insight that will help us develop a mechanistic account for structural priming in the coming chapters.

2.4 Conclusion

In this chapter, we have looked at structural priming both as an investigative tool for studying language production and as a cognitive phenomenon, itself meriting investigation. As an investigative tool, structural priming provides insight into the processes of language production, especially those related to grammatical encoding. It confirms the separation of some levels previously predicted on the basis of speech errors and also suggests further stratification of these levels. Structural priming in dialogue also shows that processes of language production are intricately related to processes of language comprehension, helping interlocutors to align their mental states with each other. As a cognitive phenomenon, structural priming lies at the heart of the debate about the nature of learning during linguistic processing. One possibility is that structural priming is a consequence of language acquisition and another possibility is that it is a transient memory trace useful during a discourse, which might or might not be used for long-term language learning. One way to resolve this debate is to look at the duration of structural priming and we have reviewed studies that show that structural priming persists for up to a lag of 10 intervening utterances, while its lexical enhancement seems to decay relatively quickly. But does this necessarily mean that structural priming is used in language acquisition? This question cannot be answered without a knowledge of the exact learning mechanisms that lead to structural priming and lexical boost. The goal of this thesis is to give one such mechanistic account. We begin by reviewing two existing accounts (one formal and one conceptual) in the next chapter and point out their successes and their pitfalls.

3 Models of Structural Priming

3.1 Introduction

This chapter discusses theoretical accounts of structural priming. Such theoretical accounts come in two flavours: computational models, which are specified using a formal mathematical framework, and conceptual models, which systematically specify the flow of information between different modules. Computational models have the advantage of being precise and can therefore be tested through simulation while conceptual models have the advantage of being more general and can have several possible mathematical implementations.

We discuss one formal computational model – the error-correction model proposed by Chang et al. (2006) – and an alternative conceptual model – the trailing-activation account. We discuss the strengths of each model and, more importantly, identify the gaps that need to be filled. Structural priming is, after all, a form of learning and so we start with some general comments about how models learn from their environment (section 3.2). We outline, through these comments, the broad distinctions between different learning paradigms. After these preliminary comments, we dive straight into the description of the error-based model, providing a very short summary and concentrating mainly on identifying the limitations of this approach (section 3.3). We point out these limitations not because we believe that an error-based approach to understanding priming is wrong, but that it is limited under certain circumstances. In section 3.4, we develop a formal theory that quantifies how well error-based learning explains structural priming under different circumstances. Finally, we briefly describe a conceptual account for structural priming – the trailing-activation account – and its precursor, the spreading-activation theory (section 3.5). As yet, this account has not

been implemented as a formal computational model. Therefore, instead of pointing out its limitations, we focus on mechanisms that need to be elaborated before this account can be expanded into a precise mathematical model that can be tested.

3.2 Learning in computational models

Scientific investigation begins with the observation of a physical system. The systematic recording of observations provides a list of behaviours that form the raw data. The goal of scientific enquiry is to understand the underlying reasons for the production of this data. The investigation expresses these reasons as a theory or a model that explains the data in terms of a set of processes operating over a group of representations. The next stage in research is to match the behaviour of the model with that of the physical system. The behaviour of the model is governed by its input and a set of free parameters. The investigation adjusts these free parameters so that the model closely matches the observed behaviour of the physical system. This adjustment of the free parameters as the model comes into contact with its environment is termed as *learning* (Mendel & McLaren, 1970).

Another way to look at a model is as a transformation of an input stimulus to an output or a response. The transformation describes the behaviour of the model and depends on a set of free parameters. As the model comes in contact with its environment, it generates a response but it also changes its parameters. The model not only transforms the input, but also gets transformed by the input. Various learning algorithms describe how computational models can both process information and perform learning at the same time. In this section we consider two learning paradigms that achieve this goal. In the rest of the chapter we look at specific learning algorithms and models developed to explain structural priming.

§ 3.2.1 Teachers and pupils

Let us consider a computational model that tries to approximate the stimulus-response characteristics of a physical system. As the model comes into contact with the environment, it would perform transformations on the input signals and generate an output. This output might or might not match the output that the physical system would have produced. If we want the model to learn from this episode of information processing, it will be useful to give the model information about the correctness of the output signal.

In other words, the model needs a *teacher*, that possesses knowledge of the physical system and can ascertain the correctness of the response for each given input. The learning paradigm that assumes the existence of such a teacher is called *supervised learning*.

During supervised learning, the model is trained on a pairs of inputs and target outputs. The training simulates the model with each input on the list and generates an output. A teacher compares this output with the target and generates an error. The model then uses this error to adjust its free parameters such that the error is reduced if the model was simulated again.

Of course, if we want to use this learning paradigm we have to assume that a teacher will be available that possesses enough knowledge about the environment and we can use this teacher to predict the target output for every given input. But this assumption might not always be true. Consider a system that models language comprehension. The target output of such a system in not entirely clear. Depending upon what stage of language comprehension we are modelling, the output might be a concept, a proposition or a thought. Because these are abstract cognitive constructs, it is difficult to know the target output of the model for any of these stages of comprehension. In the absence of this knowledge, we cannot use supervised learning for modelling language comprehension. As we shall see below, Chang et al. (2006) used a trick to overcome this difficulty. Instead of using any of these abstract cognitive constructs as output, their model used the input utterance itself as the output and a delayed version of the utterance as the input.

The alternative to supervised learning is learning without a teacher. This learning paradigm is, rather unimaginatively, called *unsupervised learning*. The goal of the model remains the same: to approximate the behaviour of a physical system. Because we do not have a teacher that can give us the target response for each input, we must use a heuristic for adjusting the free parameters of the model. The system may have an internally derived training signal based on the system's ability to to predict its own input, or it may be some more general measure of the quality of its internal representation (Becker, 1995). One of the earliest and most popular unsupervised learning algorithms is Hebbian learning, which postulates that the synaptic efficacy between pre- and post-synaptic neurons should increase whenever they get co-activated. Instead of comparing the response of the system to a target output, this algorithm learns patterns in the input stimuli.

Both supervised and unsupervised learning paradigms assume that the goal of the

model is to approximate the input-output behaviour of a physical system. However, learning might not be targeted at achieving a particular transformation from input to output, but only at changing a system in a specific way. A finite-state transducer (FST) is a good example of this form of learning. When an FST receives an input, this input (possibly) changes the state of the FST. We can say that the FST has undergone learning. But the FST does not try to approximate an input-output behaviour as a result of this learning. The output depends not only on the input, but also on the state that the FST was in, when it received the input. Declarative memory is another example. An episode of stimulation might leave a trace in declarative memory but this trace need not approximate any input-output transformation. Rather, the trace simply acts as a record of the episode.

Well known learning algorithms such as error-correction learning and self-organised learning try to optimize a global objective function. As training proceeds, the model iteratively changes its parameters so that the sequence of changes converge to an optimal value of this objective function. Learning is governed by a global variable and changes local values. We call this form of learning the *top-down* approach to learning. In contrast, we saw the example of declarative memory where the synaptic adjustment may or may not lead to a global objective function (Becker, 1995). This form of learning can be called *bottom-up* approach to learning. One of the reasons why Hebbian learning is so popular is that it combines bottom-up synaptic learning with achieving a top-down globally optimised objective function. Later in this chapter, we will present the trailing-activation account of structural priming which falls into the bottom-up approach to learning. Although we will be using unsupervised learning algorithms to implement the trailing-activation account, the reader should remember that our aim will be to record traces of information rather than internalising an input-output transformation.

3.3 Error-based learning model

It is uncontroversial that priming is a consequence of some kind of learning. The question is which part of the cognitive system undergoes learning and what is the mechanism of such learning. In Section 3.2.1 we saw that learning could be supervised or unsupervised. In this section, we consider a previous model that uses supervised learning and tries to explain priming as a consequence of such this mechanism.

§ 3.3.1 A brief summary of Chang et al. (2006)

Chang et al. (2006) presented a language-comprehension and production model that relies on error-correction to learn the sequential structure of utterances. When this model is given an input utterance, it breaks down the utterance into a sequence of words. As each word is processed by the system, it tries to predict the next word. In order to make such a prediction, the system maintains an internal model of the structure of sentences. If the prediction is incorrect, the system generates an error. This error is backpropagated through the system so that the elements of internal model that led to the incorrect prediction are penalised. Thus, for each utterance, the system makes a sequence of predictions and adjusts its internal model based on these predictions. Chang et al. (2006) showed that a system based on these principles is able to successfully extract the abstract structure of utterances so that it can produce grammatical utterances if it is given the meaning of these utterances as its input. Figure 3.3.1 shows the flow of information through this model.

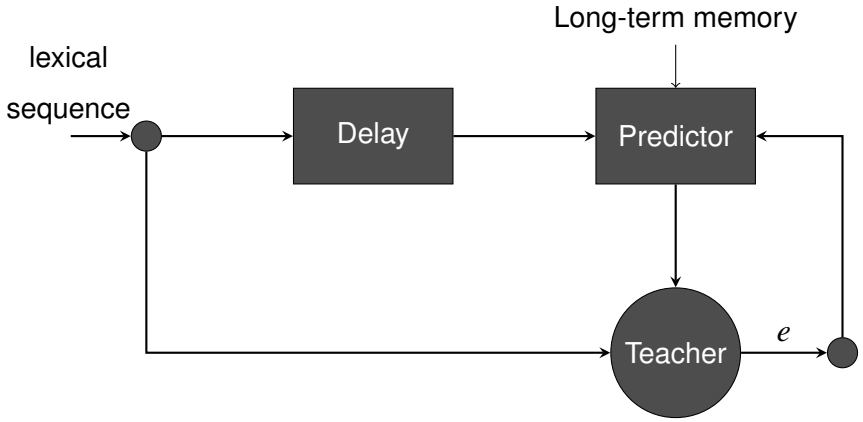


Figure 3.3.1: [Supervised comprehension] To perform comprehension using supervised-learning, Chang et al. (2006) used the lexical sequence both as the input and as the target. The predictor forecasts the next word using the given sequence of words. The teacher, then, compares this prediction with the input signal and generates an error, which is used to adjust the rules of prediction.

Internally, the model presented in Chang et al. (2006) (hereafter CDB06) consists of two parallel pathways for the flow of information. While the *meaning pathway* represents aspects of semantics that are critical to sequencing of utterances, the *sequencing pathway* consists of a sequential recurrent network (Elman, 1990) that encodes the structure of sequences themselves (Figure 3.3.2).

CDB06 implements the model using a connectionist framework. It represents words, their features, syntax and semantics as patterns of activation over units in a network. These units are connected to each other through weighted links and the system encodes its internal rules by adjusting weights on these links. The system learns most of these weights through error-backpropagation, and some of them are set manually. During comprehension and production, activation spreads in the network along these weighted connections, through both meaning and sequencing pathways. Different patterns of activation compete with each other for higher activation and the network makes a prediction based on the pattern that receives largest activation.

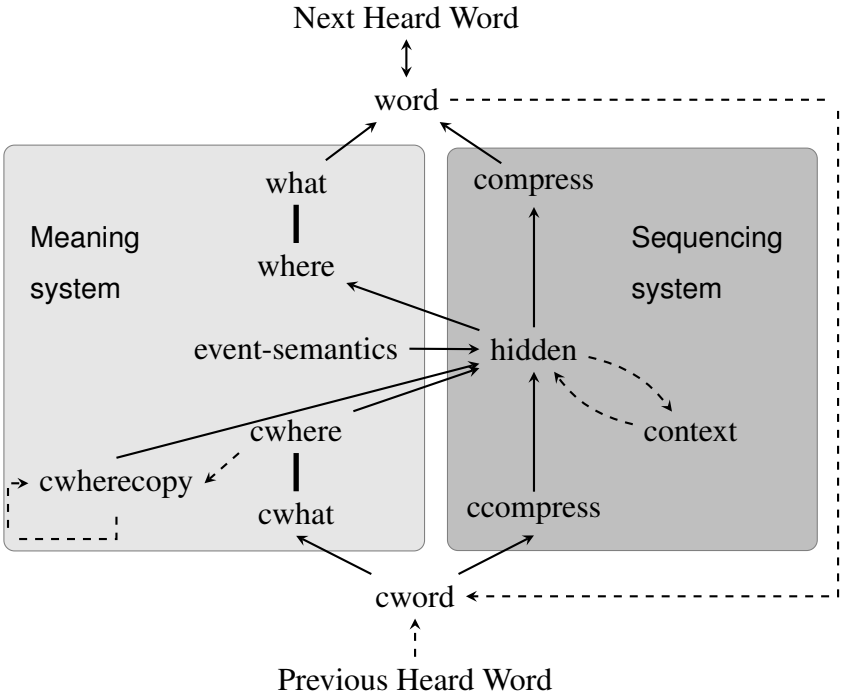


Figure 3.3.2: [Dual path model] The *Meaning system* and *Sequencing system* are parallel pathways for the flow of information in CDB06. The sequencing system performs categorisation using the *ccompress* and *compress* units and it performs sequencing using the sequential recurrent network implemented by *hidden* and *context* units. The thick lines between *what* and *where* units represents connections that are manually set at the beginning of the trial. Adapted from Chang et al. (2006).

At the heart of CDB06 is a sequential recurrent network (SRN). It is this element that allows the model to predict a word based on its context. An SRN is a feedforward network with the hidden layer connected to a recurrent layer. This recurrent layer stores the context, or the previous values, of the hidden layer. Every time the network

receives an input, the output of the hidden layer is governed not just by this input, but also by the context stored in the recurrent layer. Elman (1990) showed that this setup in an SRN gives it the capability of remembering sequences of patterns. Chang et al. (2006) trained the network on a subset of the English language. They demonstrated that, after training on this subset of language, the network was able to predict words that were in grammatically correct positions 89% of the time and utterances that were semantically correct 82% of the time.

The sequential recurrent network developed by Elman (1990) was extended in two ways by Chang et al. (2006). First, they introduced a meaning pathway that runs parallel to the SRN and connects to the hidden layer of the SRN. This meaning pathway encoded the thematic roles of words and event-semantics of utterances. As a result of these connections, the SRN started to predict sequences based not just on input words, but also on the thematic and event-semantic properties of utterances. Next, Chang et al. (2006) introduced a set of *compression* units that lie between the internal and external representations of the SRN. These compression units reduced the dimensionality of external representations. As a result of this dimensionality-reduction, the SRN based its sequencing decisions on word classes rather than individual words. As the network learnt to order utterances, it also learnt to abstract away from specific words to word-classes, thus making the system less lexically specific with training. Together, the two extensions made the network more suited for natural language production and ensured that the network predicted word-categories based on the intended message.

§ 3.3.2 Testing CDB06

Chang et al. (2006) used their network to model two cognitive phenomena: Syntactic acquisition (during development) and structural priming (during a conversation). Both acquisition and priming are forms of learning. The central claim of Chang et al. (2006) was that both these phenomena rely on the same algorithm of learning: backpropagation of error generated by comparing predicted and actual input. They claimed that the cognitive system maintains an internal model of linguistic structure which it uses to make predictions during training. When the cognitive system makes predictions that do not match the external input, it adjusts its internal model. Over a long period of time, this learning algorithm leads to an internal model that correctly predicts the external input. Crucially, Chang et al. (2006) also claimed that over a shorter period of time, the same learning algorithm led to structural priming. Priming and acquisition

are manifestations of the same higher cognitive principle: prediction-based learning.

To substantiate their claim, Chang et al. (2006) had to show that their model could reproduce experimental findings on priming and acquisition. Therefore, they needed to design simulations that replicated the procedure of experimental studies. Priming studies investigate how comprehension or production of utterances affect syntactic choices during subsequent utterances. Thus these experiments consist of *comprehension trials*, where subjects are required to understand an utterance that they see or hear, and *production trials* where the subjects are required to produce an utterance for a given picture or situation. Because Chang et al. (2006) wanted to replicate these experiments on the model, they defined corresponding comprehension and production trials for their model.

Definition of a production trial is simple. For subjects, production is the act of converting an abstract message into a sequence of words. The trial adopted this definition and initiated production by supplying the model with a message. It manually set the activation of conceptual, thematic and event-semantic units in the meaning pathway. It also set the strength of connections between these units. Given this message, the model was expected to produce a sequence of words. Since this was a production trial, the model received no external input and it did not perform any error correction.

Comprehension trials, however, are a bit more complicated to define. They always involve an external input, which is the sequence of words that the model is required to 'understand'. But this understanding can either proceed in the absence of any contextual information – i.e. without a given message – or in the presence of the intended message, in which case comprehension involves learning to match the utterance with the message. The latter kind of comprehension trials are essential if the model wanted to learn the connection strength between the meaning and sequential pathways. Chang et al. (2006) called these trials *situated events*, while the trials in which the model performed predictions in the absence of a message were called *messageless events*.

Chang et al. (2006) trained the model on a mixture of situated and messageless trials and tested it for messageless events followed by production trials. During both training and testing phases, the model learnt (through error-backpropagation) at the end of (both kinds of) comprehension trials. The amount of priming was measured as the difference between the percent of target structure produced after prime of the same and competing structures.

The trained model replicated results from Bock and Griffin (2000), showing a main effect of prime structure – i.e. the model was more likely to choose a syntactic structure

if it had recently comprehended a prime with the same structure. As in Bock and Griffin (2000), the amount of priming persisted over lags of up to ten intervening filler trials. This result verified that error-based learning leads to both short-term and long-term changes in the system.

Chang et al. (2006) also replicated several other experimental findings which demonstrated that the model shows structural priming irrespective of thematic role overlap between prime and target. Both locatives such as *The wealthy widow drove an old Mercedes to the church* and prepositional datives such as *The wealthy widow gave an old Mercedes to the church* primed a dative target to the same extent (Bock & Loebell, 1990). Since the locatives and prepositional datives have the same structural form and since the model showed same amount of priming for both structures, Chang et al. (2006) argued that the model's sequencing system generalised over the two different thematic roles, making predictions based solely on each utterance's surface structure.

Finally, Chang et al. (2006) used the same model to explain patterns of syntactic acquisition. They tried to resolve the conflict between two contrasting observations of language acquisition by proposing that the same internal principles could manifest themselves as these contrasting observations. While the *early-syntax* theory (Fisher, 2002; Gleitman, 1990; Naigles, 2002) claims that children exhibit an ability to make structural classifications at an early age, the *late-syntax* theory (Bates & Goodman, 2001; Lieven, Behrens, Speares, & Tomasello, 2003; MacWhinney, 1987; Tomasello, 2003) claims that children's initial syntactic constructions are lexically specific and that they arrive at abstract syntactic constructions only at a later age. Early-syntax theories rely on evidence from preferential-looking data in children and late-syntax theories rely on evidence from sentence production. Chang et al. (2006) attempted to resolve this debate by showing that different estimates of syntactic competence in the model can explain observations compatible with both early and late-syntax theories. One estimate, the *error-difference score*, measured the amount of preferential looking while another estimate, *grammaticality*, measured the correctness of produced utterances. By analysing the value of these estimates at different stages of training, Chang et al. (2006) demonstrated that the model agreed with the early-syntax theory when syntactic competence is measured as the error-difference score, but with the late-syntax theory when syntactic competence is measured as grammaticality.

The major success of CDB06 is to demonstrate that the same principle of error-based learning can seamlessly explain the short-term phenomenon of structural priming and the long-term data regarding language acquisition.

§ 3.3.3 Limitations and Critique

Chang et al. (2006) conjectured that the mechanics of both structural priming and language acquisition rely on the engine of prediction-based learning. Their model provides proof that prediction-based learning can indeed lead to both repetition of syntactic structure and development of abstract syntactic knowledge. However, claiming that prediction-based learning *can* drive priming and acquisition is different from claiming that it *does* drive these phenomena. Their computational model demonstrates the possibility, but it cannot prove that priming and acquisition are indeed driven by prediction-based learning. In this section, we consider at some specific limitations with the implementation of CDB06. In the next section (3.4) we will develop a theoretical account that allows us to verify whether error-based learning underlies structural priming.

§ 3.3.3.1. **Lexical enhancement of structural priming.—** As we discussed in Chapter 2, Pickering and Branigan (1998) showed that when lexemes are repeated between prime and target, the amount of structural priming increases. This result, known as the *lexical-boost effect*, shows that the lexical context of a syntactic decision plays a role in the mechanics of structural priming. There is an ongoing debate about the duration of this lexical boost (see section 2.3), but its existence does indicate a lexical element to syntactic retrieval. If subjects show structural priming when they retrieve syntactic representations, then this finding suggests that lexical context of the stored representations influences this retrieval. Crudely, the syntax module needs to be connected with the lexicon and flow of information along these connections plays a causal role in the subject's choice of structure.

Chang et al. (2006) tested their model for lexical boost by repeating the experiment conducted by Pickering and Branigan (1998) on their model. The model showed structural priming in the absence of any lexical overlap. However, they also found that the amount of structural priming in their model does not increase with lexical overlap – i.e. they did not find a lexical-boost effect.

The absence of lexical-boost in CDB06 is no coincidence. It is a consequence of the fact that the model is designed to learn from predictions. The goal of learning in CDB06 is to accurately predict the next word, given a series of words. Even with a limited vocabulary, this is a very difficult task for the model because there are always several words that are equally likely to occur at a particular position in the utterance. To get around this difficulty, the SRN in CDB06 chose not to predict the exact word

itself, but instead it predicted the part-of-speech for the next word. This prediction is a far more achievable goal. Of course, there could still be syntactic alternatives at the end of each word, but the number of such alternatives is much more limited. To achieve this part-of-speech prediction, Chang et al. (2006) surrounded the SRN by the *ccompress* and *compress* layers (Figure 3.3.2). These compression units insulated the SRN from individual words and instead allowed it to sequence word classes. This insulation of the SRN means that the lexical context of a syntactic decision was lost and the model showed no lexical-boost effect.

One could object that the SRN might have contact with the lexical context through the meaning pathway. However, this pathway has limited connectivity with the sequencing system. Specifically, only the *where* and *event-semantics* units make contact with the sequencing system. Since the lexemes themselves map only onto the conceptual *what* units, the meaning pathway too does not allow any lexical influence on structural decisions.

Chang et al. (2006) proposed that the reason why CDB06 did not replicate lexical boost was that it modelled only an implicit memory of sentence structure and that lexical boost is due to an explicit memory of the sentence. They suggest that this implicit memory consists of abstract syntactic knowledge stored without any lexical context. While this division of syntactic knowledge into explicit and implicit memory might be true, their proposal raises the question: what is the relative contribution of the two forms of memory on structural priming? The strength of CDB06 is to suggest that short-term structural priming and long-term acquisition rely on the same learning mechanism: error correction between predicted and actual input. The suggestion that both implicit and explicit memory separately contribute towards structural priming undermines the possibility that the same mechanism is responsible for both short-term (priming) and long-term (acquisition) phenomena.

Also, CDB06 does not state the connection between explicit and implicit memories. Are they separate systems or does the explicit memory of the utterance play a role in forming abstract structural representations? Eichenbaum (2003), for example, hypothesised that semantic memories could be formed by learning the relationships between episodic memories. If that were the case then it is possible that such (explicit) episodic memory contributes to structural priming while (implicit) abstractions over these episodic memories contribute to language acquisition. Such an account has a much better explanation for the relative interaction of explicit and implicit memories, and also explains how memory is consolidated and becomes more abstract. The trouble

for CDB06 is that it has no account of how such explicit memories will gradually turn into abstract syntactic knowledge. In fact, it argues that the abstract syntactic knowledge is derived, on-line and immediately during comprehension. One could rightly ask whether there is any advantage in updating the abstract syntactic knowledge on-line and, more importantly, whether subjects actually do this.

§ 3.3.3.2. **Production-to-production priming.—** Prediction-based learning is a form of supervised learning. The prediction of the model is compared with an expected target and an error is generated. The expected target is available from the external signal during comprehension. The error measures the mismatch between the prediction and the external signal. The greater the mismatch, the more the system needs to change. Therefore learning is proportional to the error. The backpropagation algorithm proves that incrementally learning to minimize the mismatch between the prediction and external input converges to minimise the error (Rumelhart, Smolensky, McClelland, & Hinton, 1986). In other words, as the training progresses the system starts making better predictions.

That an expected target is available to compare the output of the model is, of course, an assumption. In the absence of an expected target, the prediction cannot be measured for correctness and an error cannot be generated. Without generating an error, the system cannot learn. And while an expected target is available during comprehension, finding such a target is not straightforward during production. For such a target to exist, the speaker would need to compare their prediction with the output signal – i.e. the speaker would need to monitor their own speech and find a mismatch between their prediction and the external speech. Specifically, for structural priming to exist, such an error would need to be a syntactic mismatch between the external speech and their predictions. We would see below that if such a mismatch between the predicted and output syntax exists, then it would require us to make a some assumptions about the language system that are difficult to defend.

Thus, an error-based learning account has no choice but to not perform any learning during production. This is indeed the case for CDB06, which only tested a comprehension-to-production priming. While testing CDB06, Chang et al. (2006) used a *messageless* event as a prime trial and a *production* event as a target trial. A messageless event gave a sequence of words (the utterance) as the input, but did not activate the units in the meaning pathway. This sequence of words provided the expected target with which the system could compare its predictions. Since the messageless event was a form of

comprehension event, all the test cases measured comprehension-to-production priming. If, on the other hand, the model was given a production event as a prime, it will undergo no learning and show no production-to-production priming. Thus if structural priming is prediction-based, as Chang et al. (2006) argued it is, then subjects should show no production-to-production priming.

Production-to-production priming, i.e. experiments in which both the prime and the target consist of sentence production trials, is difficult to measure in the lab. Since the experiments try to measure the amount of repetition between a prime and a target, the experimenter wants to control what the subject hears or speaks during the prime trial. It is during the target trial that the subject gets to choose the syntactic structure, which the experimenter can then compare with the given syntactic structure of the prime. This means that even in experiments that, ostensibly, measure production-to-production priming (e.g. Branigan et al. (2000)), the priming trial consists of implicit comprehension which is then followed by production. Therefore, it is difficult to determine whether the priming is actually obtained from the production event itself. Indeed, Bock et al. (2007) found that the amount of priming during comprehension-to-production events was similar to (production+comprehension)-to-production, showing that production of prime did not actually lead to any increase in priming.

Even though production-to-production priming is difficult to measure in the lab, we have evidence for it from corpus analysis. Gries (2005) analysed the International Corpus of English (ICE-GB) for syntactic persistence and Szmrecsanyi (2006) did an across-corpus study involving the British National Corpus (BNC), the Corpus of Spoken American English (CSAE), the Corpus of Spoken Professional American English (CSPAE) and the Frieburg Corpus of English Dialects (FRED). Both studies found that the speaker's own utterances were a significant determinant of syntactic persistence. In other words, utterances in these corpora show a significant production-to-production priming. Furthermore, both these studies obtained marginally significant results showing that as compared to comprehension-to-production priming, production-to-production priming is stronger.

Thus results from these corpus based studies showed clear evidence of production-to-production structural priming and seem to be in conflict with the results from Bock et al. (2007), which showed no advantage of production during structural priming. How do we resolve these apparently contradictory findings? The conflict between the two results arises when we assume that because priming during comprehension-to-production and (production+comprehension)-to-production experiments is similar,

production does not contribute towards priming. This assumption relies on a further assumption that the relative contributions of comprehension and production towards structural priming during a (production+comprehension) trial should add up linearly:

$$Priming(comp + prod) = \alpha Priming(comp) + \beta Priming(prod)$$

If this relationship holds, we can conclude that an equivalence in $Priming(comp + prod)$ and $Priming(comp)$ implies that $Priming(prod)$ is close to zero. But such an assumption about linearity in human memory is unfounded. The relative contributions of comprehension and production trials could add up nonlinearly:

$$Priming(comp + prod) = S(Priming(comp), Priming(prod)).$$

where $S(\cdot)$ is a nonlinear function. If, for example, $S(\cdot)$ is a sigmoidal over the sum of the two kinds of priming, then the overall amount of priming will saturate and the equivalence between $Priming(comp + prod)$ and $Priming(comp)$ does not guarantee that $Priming(prod)$ is zero. Thus, the results from Bock et al. (2007) do not imply that there should be no production-to-production priming and the study becomes compatible with Gries (2005) and Szmrecsanyi (2006) who clearly found production-to-production priming in corpora.

The existence of this production-to-production priming is problematic for a prediction-based account. If this account wants to justify the existence of priming from production trials, it must make the hypothesis that production itself involves comprehension. There are two possible ways in which this is possible: (a) people listen to their own speech and this means that a comprehension trial automatically follows every production trial, or (b) comprehension and production trials share cognitive resources and processes. However, it is not merely sufficient to say that production is accompanied by comprehension; one also needs to state the nature of this overlap. Error-based learning takes place only when prediction and external target are mismatched. Because the mismatch cannot occur during the production trial itself, it must occur during the (implicit) comprehension trial. That is, there should be a mismatch in predictions between the comprehension trial and the signal available from the production trial. Since the comprehension trial happens as a result of listening to one's own production (or overlap in comprehension and production processes), a mismatch in predictions implies a mismatch in two consecutive predictions of the system. That is, a production trial will only show priming when the system makes contrasting predictions during two consecutive cycles.

So if we assume that structural priming is prediction-based, we must make some other assumptions to justify production-to-production priming. First, we need to assume that production implicitly involves comprehension. Next, we have to assume that priming occurs only when the prediction made during production does mismatch with the prediction made by the implicit comprehension that immediately follows the production.

Are these assumptions tenable? There is still a lot of work in progress that is trying to look at the overlap between production and comprehension (Pickering & Garrod, 2007). However, the second assumption is difficult to defend. The (implicit) comprehension immediately follows the production and there can be no learning between the two processes. In the absence of learning, the system makes the same prediction during the two consecutive processes. The same predictions means no mismatch and no mismatch means no learning. There can be only one possible cause for different predictions during consecutive cycles: Internal noise. Thus priming during production-to-production trials should be proportional to internal noise. On the other hand, priming in comprehension-to-production trials will be due to both the internal noise and a mismatch between listener's prediction and speaker's prediction. Considering that the two interlocutors are not in absolute alignment, it is fair to assume that their predictions will not always match (even in the absence of noise). Thus, comprehension-to-production trials should show a larger mismatch error and a larger priming than production-to-production trials. This conclusion contradicts both experimental observations and (Bock et al., 2007) and results from corpus analysis (Gries, 2005; Szmrecsanyi, 2006). We can conclude that implicit comprehension cannot be the reason behind production-to-production priming.

§ 3.3.3.3. **Independence of structural and semantic pathways.—** One key finding of existing studies on structural priming is that it seems to exist independent of semantic overlap between prime and target trials. Bock and Loebell (1990) have shown that subjects produce an increased number of prepositional datives such as *IBM promised a bigger computer to the Sears store* not only when they are primed using another prepositional dative such as *The wealthy widow gave an old Mercedes to the church* but also when they are primed using locatives such as *The wealthy widow drove an old Mercedes to the church*. Crucially, locatives are as good as prepositional datives for priming prepositional datives. Since locatives and prepositional datives differ in their thematic roles, Bock and Loebell (1990) argued that structural priming is inde-

pendent of thematic role overlap. They argued that this is an important finding because it showed that structural representations are independent of syntactic representations and are not identifiable with conceptual information, such as thematic roles.

Chang et al. (2006) replicated these results using their error-based model and argued that these results showed that the model successfully abstracts structural representations. Indeed, the dual-path architecture of their model would suggest that it should be able to separate sequential and semantic representations and that each of these representations should be able to independently influence structural decisions.

However, a closer look at the system presents a problem. While CDB06 had largely independent representations of sequential and meaning subsystems, it also included a limited amount of connectivity between the two subsystems. In fact, the connections from the *cwhere* and *event-semantics* units in the meaning subsystem to the *hidden* units in the sequential subsystem (Figure 3.3.2) allowed the sequencing decisions to be dependent on thematic roles and event-semantics respectively. These connections were crucial during a production episode, where the system was only given the message and had to make structural decisions based on the message. It is therefore surprising that the model included these connections between meaning and sequencing subsystems and yet showed that both locatives and prepositional dative show an equal amount of priming in spite of different conceptual representations. How can we explain this dichotomy between the model's architecture and its behaviour?

One explanation is that the model successfully generalised structural representations and since locatives and prepositional datives have the same surface structure, each construct provided the same amount of priming. This is the explanation that Chang et al. (2006) forwarded in their paper. If this hypothesis is correct, it explains why locatives are able to prime prepositional datives. However, it still does not explain why the conceptually identical prepositional dative does not prime more than the conceptually dissimilar locatives.

The answer to this second question lies in the conceptual representation of CDB06. Instead of using the traditional thematic roles of *Agent*, *Patient*, *Theme*, *Goal*, etc., CDB06 used more abstract roles *X* (agent, causer), *Y* (patient, theme) and *Z* (goal, location, benefactor). These XYZ roles represented locatives and prepositional datives identically. In the above example, *The wealthy widow* ..., X = *wealthy widow*, Y = *old Mercedes* and Z = *church* for both the locative and dative phrases. Thus it is no surprise that CDB06 showed that prepositional dative and locative show similar amount of priming since they had identical conceptual structure in the model.

In fact, the conclusion that the model has generalised the surface structure so that prepositional datives and locatives are represented identically is questionable. The real reason why locatives showed priming at all is that adjuncts (*... to the church*) are optional in locative phrases. Thus sometimes the system predicted an *end-of-sentence* after the object and when it received the adjunct phrase (*to the church*), it underwent learning. Had the adjunct phrase not been optional, the two prepositional dative and locative would not have shown similar level of structural priming, even though they had identical surface structure. Note that we are not objecting to the independence of structural learning from conceptual information, but questioning whether the SRN, with connections from the meaning system, is able to capture such an independence. The above experiment does not distinguish a model that shows similar priming for prepositional dative and locatives due to the optional adjunct phrase in locatives from one that shows similar priming based on the identical surface form of the two. In order to distinguish the two accounts, an experiment should compare the amount of priming from two sentences with same surface form but different thematic roles and, crucially, no optional parts. Such an experiment can possibly give identical priming from the two sentences in subjects but unequal priming in the model.

The second experiment in Bock and Loebell (1990) provided a stronger test for the independence of conceptual and structural representations in CDB06. This experiment found that subjects produced an increased number of passives such as *The construction worker was hit by a bulldozer* not only when they were primed using another passive, but also when they were primed using an intransitive locative such as *The construction worker was digging by the bulldozer*. Again, these results suggest that subjects represent structural information separate from conceptual information because the locatives and passives have different conceptual structure.

In this case the XYZ roles for passives were actually different from those for locatives. Therefore the model's simulation of these results did actually test whether the connections from the meaning subsystem played a role in choosing the structure of the utterance. Simulations conducted by Chang et al. (2006) replicated the results from the experiment and therefore suggested that the model successfully abstracted a structural representation that was common to the two structures. Again, it is surprising that the connections from the meaning subsystem played no part in priming.

One of the reasons is that again the XYZ representation ensured that the conceptual encoding of locatives was quite similar to that of passives, but not the same. When Chang et al. (2006) tested the model with more traditional thematic role representation,

the effect of priming did, in fact, decrease. Since the simulations fit the data better with the XYZ representations, the results in fact seem to argue that structural representation is not independent of conceptual information; rather, utterances that were traditionally thought to be represented quite differently are actually cognitively quite similar. In other words, these results argue that prepositional locatives and prepositional datives, or intransitive locatives and passives are actually semantically quite similar and it is because of this semantic similarity that they show similar amount of priming. It must be noted that this is a different point from the one Bock and Loebell (1990) were trying to make.

However, one cannot deny that the XYZ roles do not represent intransitive locatives and passives identically and yet the two utterances seem to show similar amount of priming. Thus the sequencing subsystem in CDB06 seems to base its decisions far more on the input from the word category (output of *ccompress* units) than from input from the meaning subsystem. The reason for this is the way in which the model learns through error-backpropagation after the priming trial. Error-backpropagation ensures that only those units undergo learning that play a part in producing the output. As the comprehension phase is messageless, the meaning units do not provide any contribution to the output and hence the links from meaning subsystem do not undergo any learning after the priming trial. And since there is no learning on these links, they do not play any role in priming – i.e. they bias the competing structures just as they would have done in the absence of the prime.

But this assumption of comprehension trials not involving any input from the meaning system is not correct. During a dialogue, subjects might have information about the message from the discourse context and from common ground (such as a visual scene shared amongst interlocutors). Moreover, there is no reason to expect that subjects make predictions at the structural level but do not make predictions at the semantic level. If predictive learning is performed at the structural level, as Chang et al. (2006) suggested that it is, then its also possible that a similar predictive learning is made at a semantic level. If this is true, then subjects will have (predictive) information about the message when they process the utterance. This information will ensure that learning takes place on the connections from the meaning subsystem. In order to argue that structural priming can exist independently of the conceptual information, CDB06 would have to change to a *strict* dual-path model by severing all links between the meaning and sequencing system. This, however, raises the question how such a system will lead to the grammatical production of a given message.

§ 3.3.3.4. **Lexical specificity and development of structural priming.—** Chang et al. (2006) noted that structural priming in an error-based account increases with training. They used a variable called *priming difference* to measure the amount of structural priming shown by the model. It was calculated by taking the percentage of target structures produced after the target structure prime and subtracting the percentage of targets produced after the alternative-structure prime. A model with 4,000 epochs of training showed less than 1% priming difference, while a model with 40,000 epochs of training showed around 5% of priming difference.

The reason for the increase in structural priming was the decrease in lexical specificity. When the model was young, it learnt to sequence particular words. As it went through training, it extracted categories from input words and learnt to sequence based on these categories rather than individual words. Thus a young model would learn to use the PO structure for *give* when it encounters the phrase *The sailor gave the book to the actor*, but it would *not* generalise this use to the verb *hand*. On the other hand, a relatively mature model would group *give* and *hand* together in the *ccompress* units and therefore when it increases its probability of using one verb in a PO, it automatically increases the probability of using the other verb in the same structure. Lexical specificity decreased as a result of learning and as a consequence structural priming became more robust.

But if this hypothesis is true, then children should show a larger amount of lexical specificity in structural priming as compared to adults. Such lexical specificity in structural priming can be measured through *lexical boost*, or increase in the amount of structural priming when the verb is repeated between prime and target. Contrary to this hypothesis, Branigan, McLean, Thatcher, and Jones (2006) have found that adults and children between the age of 3 and 4 years show an equal amount of lexical boost. While there is evidence from developmental studies such as Tomasello (1992) that children do show lexical specificity at an early age, this lexical specificity does not appear in structural priming experiments. Together, these two results would suggest that structural priming needs to be separated from the process of extracting abstract syntactic structure from word forms. For if structural priming is not conflated with structural acquisition then it is possible that priming shows similar patterns in children and adults in spite of differing abilities of structural abstraction.

3.4 Quantifying priming in error-based models

Chang et al. (2006) successfully demonstrate that a system designed to perform language acquisition can also lead to structural priming. But why is this the case? At one level the answer is quite simple. The same system that models the environment performs structural decisions. As the system learns, it *incrementally* adjusts this internal model. And since this model makes structural decisions, adjustment to the model leads to priming.

But is this the entire story? If an error-based account is responsible for priming, then it should not only show priming, but also explain how priming varies for different stimuli. In particular, how much priming would we obtain from stimuli of different frequencies? If we can precisely quantify the amount of priming such a model should give under different patterns of stimuli, we can compare these predictions with experimental observations. Although priming has been extensively studied, to our knowledge no account exists that systematically predicts the amount of priming under different patterns of stimuli. In the following discussion, we develop a theoretical account for priming predicted by an error-based account and discuss the experimental data that can test for these predictions. Some of this experimental data already exists, while other needs to be collected in the future.

§ 3.4.1 A theory for error-based priming

Here is the problem: assuming that priming is a consequence of error-based learning, how much priming should the system show for a given stimulus? Because an error-based model is sensitive to the frequency of the stimulus – low frequency stimuli lead to larger error – the answer to this question depends upon the frequency of the stimulus during training. In this section, we will develop a formal relation between the amount of priming shown by an error-based model and the frequency of different stimuli during training. If this amount of priming is comparable with experimental observations, we can conclude that priming could be due to error-based learning. On the other hand, if priming is larger than predicted by this relation, then we need to look elsewhere for the mechanisms responsible for this difference.

Consider an error-based system that has undergone some training and receives a test trial. The amount of priming from such a test trial is what we want to quantify. Obviously, the nature of the test trial will determine the amount of priming. But this is

something we do not know, since we are interested in developing a general account for every kind of trial. While we do not know this test trial, we will know the *probability* of occurrence of different test cases. This probability is available during the design of the experiment. Consider, for example, test cases in which the system has to choose between prepositional datives (PO) and double object datives (DO) after being trained on both POs and DOs. While we do not know whether a particular test case will be a PO or DO utterance, we would know the probability of occurrence of POs and DOs in the experiment. Thus we can develop our account of priming based on these probabilities of different test cases, or stimuli. Let us start with the fundamental premise of error-based learning:

$$\textit{Priming} \quad \propto \quad \text{Error correction } (\mathcal{E}) \text{ during trial}$$

where $\propto$ is the symbol for directly proportional. Therefore, in order to determine priming, we must determine the error $\mathcal{E}$ during a trial. This error relies on the outcome of the trial – i.e. on whether the prediction of the model matches the test case. As we stated above, we will not know the outcome of each trial in advance, so instead of considering the exact outcomes, we have to start considering their probability:

$$\begin{aligned}
\mathcal{E} &= \text{Probability of making error during trial } (\mathtt{p}^{\mathtt{err}}) \times \text{Learning rate} \\
\mathtt{p}^{\mathtt{err}} &= \text{Probability of classifying as PO when target is DO } (\mathtt{p}^{\mathtt{err}}_{\mathtt{PO}}) \\
&\quad + \text{Probability of classifying as DO when target is PO } (\mathtt{p}^{\mathtt{err}}_{\mathtt{DO}}) \\
\mathtt{p}^{\mathtt{err}}_{\mathtt{PO}} &= \text{Joint probability (Classification = PO; Target = DO)}
\end{aligned}$$

How can we evaluate this joint probability? Consider the incomplete utterance *The boxer showed* . . . . When a prediction-based model is given this phrase, it could predict that the rest of the utterance has an NP-PP structure or that it has a NP-NP structure. In the former case, the model would have classified the utterance as a PO and in the latter as a DO. The above analysis shows that priming depends upon the joint probability of the classification made by the model and the probability of the (opposite) target. In a real dialogue, the classification of a phrase by the listener (here, the model) is not causally related to the classification by the speaker (here, the target) – they are independent events. In CDB06 as well the two are independent because the model assumes prime trials to be *messageless* events – i.e. it classifies the input in absence of any semantic knowledge. From probability theory we know that the joint probability of independent events is the product of the marginals. Thus, the joint probability of

classifying the phrase as PO and the target being DO can be given by:

$$\mathtt{p}(\mathtt{c}=\mathtt{PO},\mathtt{t}=\mathtt{DO}) \quad = \quad \mathtt{p}(\mathtt{c}=\mathtt{PO})\cdot\mathtt{p}(\mathtt{t}=\mathtt{DO})$$

Here we come to the crucial question of determining the probability of classifying a particular trial as a PO: $\mathtt{p}(\mathtt{c}=\mathtt{PO})$. A neural network like CDB06 maps an input and a context onto one of the output classes. We can assume that the context and the input for both the PO and DO phrase is the same up until the choice point. Therefore, the classification will completely depend upon the weights – i.e. the training of the network. Again, we do not know how the network was trained for each simulation. But since we are developing a general theory, we do not care about the specific simulation. We would like to describe the classification probability in terms of the probabilities of different trials during training.

Let $\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}}$ be the probability of a PO trial during training and $\mathtt{p}_{\mathtt{DO}}^{\mathtt{train}}$ be the probability of a DO trial during training. Since we are only interested in the prediction of PO relative to DO, we can ignore cases in which the network is trained on other cases. In other words, $\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}}+\mathtt{p}_{\mathtt{DO}}^{\mathtt{train}}=1$. Since the network's classification depends completely on the training of the network, the probability of classification as a PO will be some monotonic function, $\mathcal{F}$, of the probability of getting a PO during training, $\mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}})$.[1] We can interpret the value $\mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}})$ as the *baseline* bias towards producing a PO. Thus, we can say:

$$\mathtt{p}(\mathtt{c}=\mathtt{PO},\mathtt{t}=\mathtt{DO}) \quad = \quad \mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{DO}}^{\mathtt{test}} \tag{3.4.1}$$

Similarly,

$$\begin{aligned}\mathtt{p}(\mathtt{c}=\mathtt{DO},\mathtt{t}=\mathtt{PO}) \quad &= \quad \mathtt{p}(\mathtt{c}=\mathtt{DO})\cdot\mathtt{p}(\mathtt{t}=\mathtt{PO})\\ &= \quad \mathcal{F}(\mathtt{p}_{\mathtt{DO}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{PO}}^{\mathtt{test}}\end{aligned} \tag{3.4.2}$$

Using equations 3.4.1 and 3.4.2, we can write:

$$\begin{aligned}\mathtt{p}^{\mathtt{err}} \quad &= \quad \mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{DO}}^{\mathtt{test}}+\mathcal{F}(\mathtt{p}_{\mathtt{DO}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{PO}}^{\mathtt{test}}\\ &= \quad \mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}})\cdot(1-\mathtt{p}_{\mathtt{PO}}^{\mathtt{test}})+\mathcal{F}(\mathtt{p}_{\mathtt{DO}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{PO}}^{\mathtt{test}}\\ &= \quad \mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}})-\mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{PO}}^{\mathtt{test}}+\mathcal{F}(\mathtt{p}_{\mathtt{DO}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{PO}}^{\mathtt{test}}\\ &= \quad \mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}})+\mathtt{p}_{\mathtt{PO}}^{\mathtt{test}}(\mathcal{F}(\mathtt{p}_{\mathtt{DO}}^{\mathtt{train}})-\mathcal{F}(\mathtt{p}_{\mathtt{PO}}^{\mathtt{train}}))\end{aligned} \tag{3.4.3}$$

[1] Note that we require the function to be monotonically increasing if the system is more likely to choose higher frequency structures over lower frequency structures.

where we have used the fact that $p_{PO}^{test} + p_{DO}^{test} = 1$ – i.e. only two outcomes are possible during testing phase: either the utterance is a prepositional object dative, or it is a double object dative.

Using Equation 3.4.3 we can express the error generated during a testing trial in terms of the relative frequencies of stimuli during the testing trials (p_{PO}^{test}) and the relative frequencies of the two structures during training (p_{PO}^{train} and p_{DO}^{train}). Figure 3.4.1 uses equation 3.4.3 to plot the probability of error during a test trial (p^{err}) against the frequency of POs during test trials (p_{PO}^{test}). Since the amount of priming is proportional to the error $\mathcal{E}$, this figure gives us a theoretical account of priming for different patterns of stimuli in an error-based model, which is what we had set out to achieve.

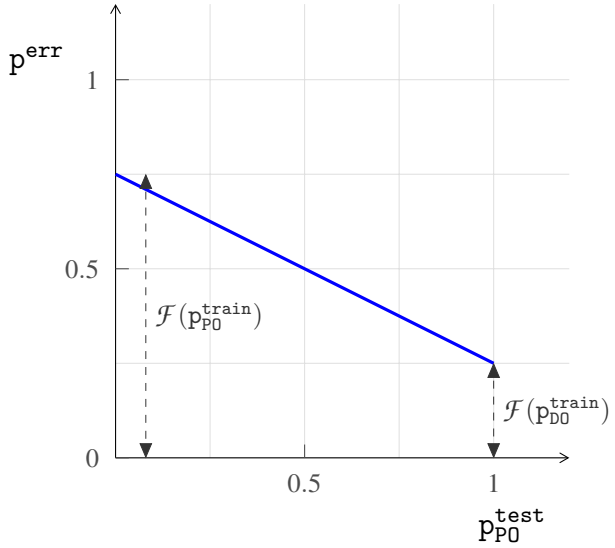


Figure 3.4.1: This plot shows the amount of priming (proportional to p^{err}) for different frequencies of PO during an experiment, p_{PO}^{test}. We can see that priming is inversely proportional to p_{PO}^{test}. This is because PO is the more frequent structure during training: $\mathcal{F}(p_{PO}^{train}) > \mathcal{F}(p_{DO}^{train})$

Now let us evaluate how a trained error-based model would respond to different frequencies of stimuli. In Figure 3.4.1, the frequency of stimuli varies along the x-axis. Because each stimulus consists of one of two constructs, PO or DO, we can represent patterns in the stimuli by using only one dimension. As the frequency of one construct increases, that of the other decreases. In Figure 3.4.1, the probability of getting a PO construct increases as one moves along the x-axis. At $(0,0)$ all test stimuli consist of DO phrases and the error is governed by $\mathcal{F}(p_{PO}^{train})$. On the other hand, at $(1,0)$ all test stimuli consist of PO phrases and the error is governed by $\mathcal{F}(p_{DO}^{train})$.

§ 3.4.1.1. **The inverse-preference effect.—** Let us first consider test-stimuli in which the lower frequency pattern is more likely than the higher frequency pattern. In Figure 3.4.1, the training data has a larger frequency of PO structure than the DO structure. This is reflected in the fact that $\mathcal{F}(\mathrm{p}_{\mathtt{PO}}^{\mathtt{train}}) > \mathcal{F}(\mathrm{p}_{\mathtt{DO}}^{\mathtt{train}})$. Now, if the lower frequency structure is more likely than the higher frequency structure then $\mathrm{p}_{\mathtt{DO}}^{\mathtt{test}} > \mathrm{p}_{\mathtt{PO}}^{\mathtt{test}}$. In other words, $\mathrm{p}_{\mathtt{PO}}^{\mathtt{test}} < 0.5$. We can observe from the above figure that in this case, $\mathrm{p}^{\mathtt{err}}$ will be larger than 0.5. Since error is directly related to the amount of priming, we can conclude that in this case the amount of priming will be above the average.

Similarly, when we test the model for the inverse pattern – higher frequency structure being more likely – the theory predicts that the amount of priming will be below the average. These two results together are the hypothesized *inverse-preference effect* (Ferreira & Bock, 2006; Chang et al., 2006; Ferreira, 2005) which predicts that the effectiveness of priming is inverse of the frequency of the structures being primed; highly frequent structures yield low priming whereas uncommon structures yield high priming.

The error-based model certainly predicts an inverse-preference effect, but do we actually observe such an inverse preference? More importantly, if we do observe this effect, can we be certain that it is due to error-based learning? We can explain the inverse-preference effect in at least two different ways: (a) it could indeed be due to error-based learning, or (b) it could be a result of attentional mechanisms since a low-frequency structure might require larger attention and lead to better encoding. The observation of inverse-preference effect and its underlying causes are still under investigation. If we do observe the inverse-preference effect we would like to ascertain that it is due to error-based learning and not due to other (e.g. attentional) mechanisms. For this, we need to look at the predictions that an error-based account will make for such an effect.

One study that is repeatedly referenced (Ferreira & Bock, 2006; Chang et al., 2006) in the context for inverse-preference effect is Hartsuiker and Westenberg (2000). This study measured the amount of structural priming in Dutch phrases ending in one of two alternative structures – an *auxiliary* or a *participle*. Besides measuring the amount of priming, Hartsuiker and Westenberg (2000) also measured the baseline levels of structural choices at the beginning and end of the experiment. They found statistically significant levels of structural priming – i.e. subjects were more likely to respond with an auxiliary-final phrase when given an auxiliary-final prime and vice versa. They

obtained another interesting result. They observed that the baseline level of priming changed after the subjects had undergone the experiment, leading them to conclude that priming accumulates over a sequence of trials rather than being a short-term effect. Their results are reproduced in Table 3.1.

Condition	Auxiliary-final	Participle-final	Other
Baseline 1	35%	56%	9%
Auxiliary-final	46%	39%	15%
Participle-final	35%	54%	11%
Baseline 2	44%	47%	9%

Table 3.1: Columns show the percent of auxiliary-final, participle-final and other responses under four different conditions. Each experiment consisted of 24 primes and 24 targets. The first six prime/target combinations constituted baseline 1, the next twelve constituted experimental trials and the last six constituted baseline 2. Adapted from Hartsuiker and Westenberg (2000).

The crucial aspect of the results in Table 3.1 that is used as evidence for inverse-preference effect is the comparison of the amount of priming under participle-final and auxiliary-final conditions, relative to the baseline. Table 3.1 shows that the number of auxiliary-final responses increases from 35% under baseline condition to 46% under auxiliary-final condition. However, the participle-final responses change from 56% under baseline condition to 54% under participle-final condition. Since the less preferred (auxiliary-final) structure shows a larger increase in responses than the more-preferred (participle-final) structure, this is seen as an evidence of inverse-preference effect. Ferreira and Bock (2006), for example, interpret that these results show that the less-preferred structure accumulates larger amount of priming than the more preferred structure. [2]

§ 3.4.1.2. **How priming changes.—** We would like to compare the results of Hartsuiker and Westenberg (2000) with the predictions of an error-based account for the given stimuli. But we can see that these results do not just talk about the amount of priming, but also how this priming changes during the experiment. Equation 3.4.3, on

[2]It must be mentioned that Hartsuiker and Westenberg (2000) themselves do not interpret these findings as amounting to an inverse-preference. Instead, they make the more conservative deduction that priming is cumulative and long-lasting.

the other hand, only predicts the priming shown by the model at the end of training. Therefore, in order to compare the predictions of our theory to these results, we must generalise our theory in two ways: first we must change the nomenclature so that we can talk about *auxiliary-final* and *participle-final* structures and not just PO and DO; secondly, we must extend the theory to see how the predictions would change after each trial – i.e. we must build in dynamics in the prediction of error.

The first goal is quite simple and so we tackle the aspect of generalising the nomenclature first. Each of equations 3.4.1, 3.4.2 and 3.4.3 are specific to PO and DO constructions. Instead, we adopt the more general names for alternative constructions: $\mathtt{x}$ and $\mathtt{y}$, giving:

$$\mathtt{p}(\mathtt{c}=\mathtt{x},\mathtt{t}=\mathtt{y}) \quad = \quad \mathcal{F}(\mathtt{p}_{\mathtt{x}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{y}}^{\mathtt{test}} \tag{3.4.4}$$

$$\mathtt{p}(\mathtt{c}=\mathtt{y},\mathtt{t}=\mathtt{x}) \quad = \quad \mathcal{F}(\mathtt{p}_{\mathtt{y}}^{\mathtt{train}})\cdot\mathtt{p}_{\mathtt{x}}^{\mathtt{test}} \tag{3.4.5}$$

$$\mathtt{p}^{\mathtt{err}} \quad = \quad \mathcal{F}(\mathtt{p}_{\mathtt{x}}^{\mathtt{train}})+\mathtt{p}_{\mathtt{x}}^{\mathtt{test}}(\mathcal{F}(\mathtt{p}_{\mathtt{y}}^{\mathtt{train}})-\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{\mathtt{train}})) \tag{3.4.6}$$

The second extension is less straightforward. We would like to determine how each test trial changes the model. As is evident from both equation 3.4.6 and Figure 3.4.1, $\mathtt{p}^{\mathtt{err}}$ depends on two parameters: $\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{\mathtt{train}})$ and $\mathcal{F}(\mathtt{p}_{\mathtt{y}}^{\mathtt{train}})$. These parameters are sufficient to describe the value of $\mathtt{p}^{\mathtt{err}}$ for different values of $\mathtt{p}_{\mathtt{x}}^{\mathtt{test}}$. As a result of error-based learning, these two parameters will undergo change. Therefore, to understand how error-based learning changes the system, we need to describe the dynamics of these two parameters.

The system undergoes learning at the end of each trial. This learning will be proportional to the learning rate, λ, and to the prediction-error. The parameter $\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{\mathtt{train}})$ describes the probability of classifying the construction as $\mathtt{x}$ based on the training. As a result of the trial, this parameter will change. How it changes will depend on the prediction-error during the previous trial. If the system predicted $\mathtt{x}$ but the target was $\mathtt{y}$, the probability of the system classifying an identical trial as $\mathtt{x}$ should be lower and correspondingly $\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{\mathtt{train}})$ must decrease. On the other hand, if the system predicted $\mathtt{y}$ but the target was $\mathtt{x}$, then $\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{\mathtt{train}})$ should increase. While we do not know the exact outcome of the trial, we do know the probabilities of the two errors, $\mathtt{p}_{\mathtt{x}}^{\mathtt{err}}$ and $\mathtt{p}_{\mathtt{y}}^{\mathtt{err}}$ (equations 3.4.4 and 3.4.5). The learning in the system should be proportional to the difference in these two errors. Thus we can write:

$$\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{(2)}) \quad = \quad \mathcal{F}(\mathtt{p}_{\mathtt{x}}^{(1)}) \; - \; \lambda\,(\mathtt{p}_{\mathtt{x}}^{\mathtt{err}}-\mathtt{p}_{\mathtt{y}}^{\mathtt{err}}) \tag{3.4.7}$$

where we have used the superscripts (1) and (2) to refer to the number of the iteration. If we set $\mathcal{F}(p_x^{(1)})$ to the initial condition $\mathcal{F}(p_x^{train})$, then we can investigate the effect of learning on the trained model by observing how this parameter changes. Using Equation 3.4.4 and the fact that $p_x^{test} = 1 - p_y^{test}$, we obtain:

$$\begin{aligned}
\mathcal{F}(p_x^{(2)}) &= \mathcal{F}(p_x^{(1)}) - \lambda\,[\mathcal{F}(p_x^{(1)}) \cdot p_y^{test} - \mathcal{F}(p_y^{(1)}) \cdot p_x^{test}] \\
&= \mathcal{F}(p_x^{(1)}) - \lambda\,[\mathcal{F}(p_x^{(1)}) \cdot (1 - p_x^{test}) - (1 - \mathcal{F}(p_x^{(1)})) \cdot p_x^{test}] \\
&= \mathcal{F}(p_x^{(1)}) - \lambda\,[\mathcal{F}(p_x^{(1)}) - p_x^{test}]
\end{aligned}$$

Using induction, we can generalise the result to the $(n)^{th}$ iteration:

$$\mathcal{F}(p_x^{(n)}) = \mathcal{F}(p_x^{(n-1)}) - \lambda\,[\mathcal{F}(p_x^{(n-1)}) - p_x^{test}] \tag{3.4.8}$$

Equation 3.4.8 gives a recursive way of finding the value of parameter $\mathcal{F}(p_x)$ at time-step (n) in terms of its value at time-step $(n-1)$. Thus it meets our objective of finding a way to describe how the model changes its predictions as a consequence of each trial. We argued above that the prediction of the model, p^{err} depends only on two parameters: $\mathcal{F}(p_x)$ and $\mathcal{F}(p_y)$. Equation 3.4.8 gives us a way of calculating both the first and, *mutatis-mutandis*, second parameters at different stages of testing.

Before putting this extended theory to test, let us see how an error-based system is predicted to evolve by this theory. Equation 3.4.8 describes the dynamics of an error-based model and as with any dynamical system, we are interested in how the dynamics will stabilise. In this case, a stable solution will tell us how the system's predictions (p^{err}) will change as the system undergoes training. We can gain a better understanding of the importance of stable solution by looking at its geometrical interpretation. As training progresses, the slope and intercept of the graph in Figure 3.4.1 will keep changing. A stable solution will tell us the shape that this graph approaches as the experiment goes on.

The recursive Equation 3.4.8 reaches stability when $\mathcal{F}(p_x^{(n)}) = \mathcal{F}(p_x^{(n-1)})$ – i.e. $\mathcal{F}(p_x)$ stops changing. This condition, in turn, implies that:

$$\begin{aligned}
&\lambda\,[\mathcal{F}(p_x^{(n)}) - p_x^{test}] = 0 \\
\Rightarrow\quad &\mathcal{F}(p_x^{(n)}) = p_x^{test}
\end{aligned}$$

In other words, the effect of error-based learning is to align the probability of classification of an input as x with the probability of the occurrence of the structure x in the test data. If we take the example of the structural choices PO and DO, then the effect

of error-based learning is to align the model's prediction of PO with the probability of occurrence of PO in the (test) environment. This result makes sense since the goal of error-based learning is to internalise the statistical structure of the environment. Thus we can understand equation 3.4.8 as the formalisation of how the model iteratively aligns its internal representations with that of the environment.

Now, let us go back to the data from Hartsuiker and Westenberg (2000) and see how well the predictions made by this theory fit their results. But first we must align the variables in the theory with values from the data. The results in Table 3.1 give us the baseline level of auxiliary-final and participle-final choices at the beginning and end of the experiment. Each of these values gives the probability that the system will choose the corresponding structure and therefore determines the parameters $\mathcal{F}(\mathtt{p}^{(\mathtt{i})}_{\mathtt{auxiliary}})$ and $\mathcal{F}(\mathtt{p}^{(\mathtt{i})}_{\mathtt{participle}})$, for a generic iteration number $(\mathtt{i})$. These two parameters allow us to determine the value of $\mathtt{p}^{\mathtt{err}}$ at the beginning and end of the experiment.

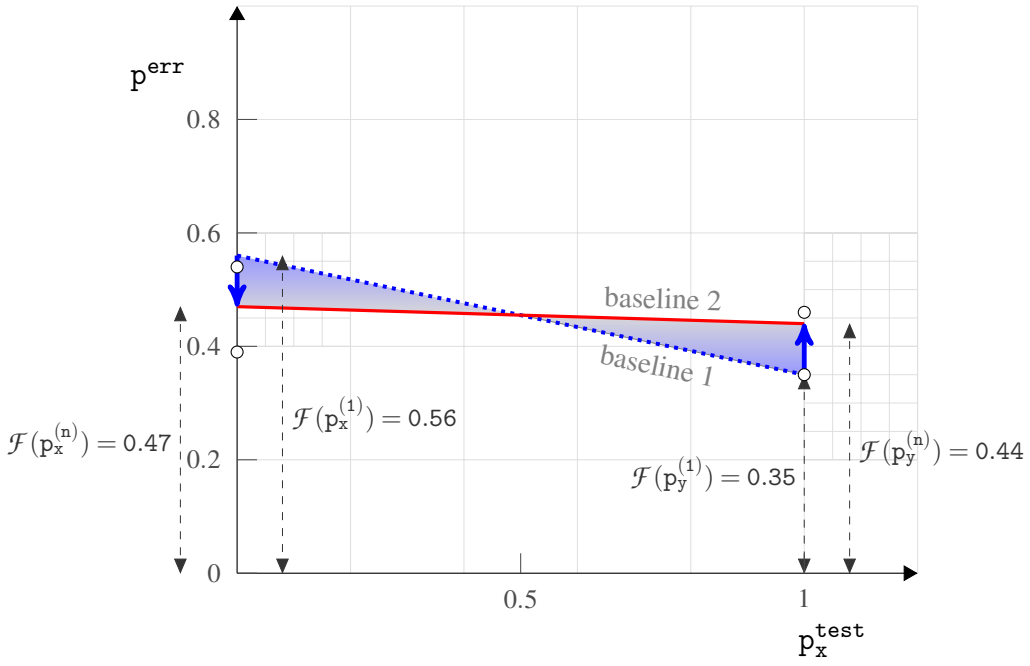


Figure 3.4.2: [Baselines] This figure plots $\mathtt{p}^{\mathtt{err}}$ for the two baseline conditions reported in Table 3.1. The variable $\mathtt{x}$ stands for participle-final construction and $\mathtt{y}$ for auxiliary-final construction. The response frequencies from the experimental trials are also marked using the small circles.

Figure 3.4.2 plots $\mathtt{p}^{\mathtt{err}}$ for the two baseline level of classifications from Table 3.1. We can observe from the figure that as baseline level of priming change the plot for $\mathtt{p}^{\mathtt{err}}$ also changes. These plots are shown as the dotted line (labelled 'baseline 1')

at the beginning of the experiment and as the solid line (labelled 'baseline 2') at the end. Since $\mathtt{p}^{\mathtt{err}}$ tells us the amount of priming that the system will undergo, we can observe from figure 3.4.2 that the system shows a larger difference in priming for the two structures at the beginning of the experiment than at the end. Formally, $|\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{(\mathtt{1})}) - \mathcal{F}(\mathtt{p}_{\mathtt{y}}^{(\mathtt{1})})| > |\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{(\mathtt{n})}) - \mathcal{F}(\mathtt{p}_{\mathtt{y}}^{(\mathtt{n})})|$, where we have used the variables 'x' for participle-final structure and 'y' for auxiliary-final structure.

As the experiment progresses, the error-based model gradually adjusts its internal representations. Before the experiment, the system believes that it is more likely to encounter a participle-final structure as compared to an auxiliary-final structure and at the end it believes that both structures are almost equally likely. This transition in the model's behaviour is shown by the thick blue line in Figure 3.4.2 (➞). The dotted line of baseline 1 slowly evolves to the solid line of baseline 2 and the model gradually changes its behaviour. In between the two lies the behaviour of the model during the experiment (shown as the shaded region). If we want to compare our prediction with the results from Hartsuiker and Westenberg (2000), this is the region to look at.

Provided the learning rate is not too large (and we argue below that it does indeed need to be small) our theory predicts that the response of an error-based system will lie somewhere inside the shaded region bounded by the plots of the two baselines. Formally, $\mathcal{F}(\mathtt{p}_{\mathtt{x,y}}^{(\mathtt{1})}) \geqslant \mathcal{F}(\mathtt{p}_{\mathtt{x,y}}^{(\mathtt{i})}) \geqslant \mathcal{F}(\mathtt{p}_{\mathtt{x,y}}^{(\mathtt{n})})$, where we have used $(\mathtt{i})$ to stand for an experimental trial that lies between the two baseline tests. For example, the probability that the model will choose the participle structure at any given point during the experiment should be between 0.56 ($\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{(\mathtt{1})})$)and 0.47 ($\mathcal{F}(\mathtt{p}_{\mathtt{x}}^{(\mathtt{n})})$).

We can now compare this prediction of the model with the results obtained by Hartsuiker and Westenberg (2000). From Table 3.1 we can see that while the participle-final responses following a participle-final prime lie within this bound (0.54), the participle-final responses following an auxiliary-final prime do not (0.39). The auxiliary-final responses too seem to be a poor fit to the data: the probability of the responses should lie between 0.35 and 0.44. While the responses following the participle-final structure are just within the bound (0.35), the responses following an auxiliary-final structure are not (0.46).

These results illustrate how our theoretical account can be used to check how well error-based models fit the patterns of structural priming. We have taken Hartsuiker and Westenberg (2000) as an example to show that if structural priming was the result of error-based learning, then the amount of structural priming would need to be bound within certain limits. Similar analyses can be performed for other studies such as

Scheepers (2003) and Ferreira (2005) to check whether the inverse-preference effect reported by such studies can be a consequence of error-based learning.

§ 3.4.1.3. **Tug of war.—** The limitation of the error-based model lies in its requirement for long-term generalization, which requires a small learning rate. If we examine the results in Table 3.1 closely, we can see that priming has a large short-term effect. The number of participle-final responses, for example, changes from 39% to 54% when its followed by a matching prime. The error-based model cannot account for such a large effect unless its learning rate (λ) is quite large. But such a large learning rate will mean that, in the long-term, learning does not converge (Oja, 1982) which would mean that the model would fail to internalise the statistics of the environment. Since Chang et al. (2006) require their model to perform language acquisition, it is critical that their model shows convergence in learning. In fact, the data from Hartsuiker and Westenberg (2000) itself shows that the baseline converges towards the statistics of the test cases – 50% for each structure. If the learning rate was large, then the baseline would shift quite radically after each priming trial making it depend largely on the most recent trial. But this does not seem to be the case. Baseline 2 seems to reflect the overall statistics of test cases presented during the experiment.

If the learning rate was large, it would also be difficult to explain why the amount of repetition after the participle-final prime (54%) is not larger than the initial baseline level of participle-final responses (56%). Hartsuiker and Westenberg (2000) suggested that the participle-final responses did not increase during the experiment because the baseline level of these responses steadily decreased as the experiment progresses. If we assume that the learning rate is large, the baseline levels will not show a steady shift and the responses would largely depend on the prime and not on the baseline. Thus, this assumption would make it very difficult to explain why the repetition after participle-final primes remains so low.

Our argument here is not that the error-based model presented by Chang et al. (2006) cannot show priming. Indeed, Chang et al. (2006) demonstrated that the model is more likely to repeat a syntactic structure than to pick the alternate structure. Since the model learnt based on error, this is not surprising. Our argument is that the amount of priming shown by the model is not sufficient to explain both the short-term and long-term effects of priming simultaneously. If the model shows priming comparable to the experiment, its baseline will shift quickly and it will not show generalisation. On the other hand, if the model shows generalisation it will show only a small amount

of priming. Thus there is a trade-off between priming and generalisation.

Similarly, priming and inverse-preference also seem to show a trade-off. The error-based model predicts larger priming if the input consists of lower-frequency structure. This can be seen in Figure 3.4.3, which plots the prediction error when one structure is highly preferred over the other.

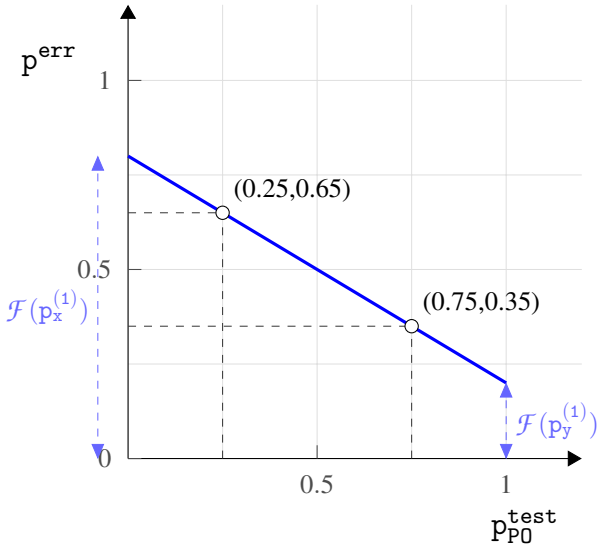


Figure 3.4.3: [Inverse preference] Low frequency primes lead to larger error correction. Mark on left shows low frequency, high priming. Mark on right shows high frequency, low priming.

We can observe from the figure that the prediction error is much larger if the probability, p_x^{test} is low. Since $\mathcal{F}(p_x^{(train)}) > \mathcal{F}(p_y^{(train)})$, we can also observe that x is the more frequent structure during training. These two observations imply that if the probability of the less frequent structure is large in the test trial, then the error-based model will show larger prediction error and consequently larger priming.

While this observation is true for the first testing trial, the behaviour of the system for subsequent trials depends on how the system changes as the result of the first trial. According to Equation 3.4.8, the shape of the plot for the second iteration will depend not only on the difference in training and test probabilities ($\mathcal{F}(p_x^{(n-1)}) - p_x^{test}$), but also on the learning rate, λ. If the learning rate is large, the system will show a large adjustment, leading to a large amount of priming. But the adjustment would also mean that the difference between $\mathcal{F}(p_x^{(2)})$ and $\mathcal{F}(p_y^{(2)})$ is diminished leading to a reduction in the inverse-preference effect. Thus, if the system is to show inverse-preference effect over a series of trials then the priming needs to be low and vice-versa.

§ 3.4.1.4. **What is inverse-preference?.—** There is one other question that needs to be addressed. So far we have assumed that the term inverse-preference refers to a larger *long-term* priming for the less preferred structure. The results from Hartsuiker and Westenberg (2000) are frequently cited (Ferreira & Bock, 2006; Chang et al., 2006; Scheepers, 2003) as evidence for such an inverse-preference effect. Ferreira and Bock (2006), for example, observed that in the results obtained by Hartsuiker and Westenberg (2000), the number of less preferred, auxiliary-final responses, seemed to increase substantially relative to the initial baseline while the more-preferred, participle-final responses, seemed to remain the same. They argued that this result suggested that

> . . . in the course of the experiment, the less-preferred auxiliary-final order accumulated more long-term priming than the more-preferred participle-final order. (Ferreira & Bock, 2006)

drawing the conclusion that inverse-preference effects are long-lived. However, we believe that this is an incorrect interpretation of the results shown by Hartsuiker and Westenberg (2000). In their study, the long-term learning shown by each syntactic structure can be measured by the change in the amount of shift in the baseline. The data in Table 3.1 shows that the amount of shift in baseline for the participle-final structure and auxiliary-final structure is the same (9%). Thus both structures seem to accumulate the same amount of long-term priming. The reason for an apparent lack of increase in priming for participle-final structure relative to baseline 1, is that the baseline level for participle-final responses keeps decreasing during the course of the experiment. In fact, Hartsuiker and Westenberg (2000) actually pointed out to this reason in their discussion of experimental results:

> Because the number of responses with the least preferred structure increases in the course of the experiment, the number of responses with the alternative structure decreases. Thus, the apparent lack of priming for participle-final responses can be explained as the result of an overall diminishing preference for the participle-final structure in the course of the experiment.

Thus, the "diminishing-preference" of the participle-final responses was not due to a greater amount of learning for this less-preferred structure, but due to an internal shift in the baseline.

In fact, if the study did show a larger long-term priming for the less-preferred structure, it would be a problem for the error-based model. The error-based theory predicts an inverse-preference effect only when the probability of the lower-frequency structure

is high during the experiment. If the probability of the two structures is the same – i.e. $p_{PO}^{test} = 0.5$ in figure 3.4.3 – then this theory predicts that the data should *not* show any inverse preference effect. From equation 3.4.6, we can see through simple algebraic manipulation that when $p_{x}^{test} = 0.5$, the prediction error becomes a constant (0.5) and does not depend any more on $\mathcal{F}(p_{x}^{train})$. The experiment design of Hartsuiker and Westenberg (2000) assumed an equal probability for the auxiliary-final and participle-final priming trials. As such, existence of a larger long-term learning for one structure would be incompatible with the error-based theory.

But the inverse-preference effect does not necessarily need to be a long-term effect. Scheepers (2003), for example, observed that the short-term priming of a low-frequency structure is larger compared to that of a high-frequency structure. When they manipulated the baseline such that the two structures reversed in frequency, they observed that the amount of short-term priming obtained for the two structures also reversed. This pattern of priming is compatible with the error-based theory which predicts that the short-term adjustment to the model as a result of a low-frequency prime will be larger as compared to a high-frequency prime, even though each structure will accumulate the same amount of long-term priming. However, if the error-based theory has to explain this short-term inverse-preference effect, then it must deal with the trade-off between inverse-preference and structural priming that we demonstrated above – i.e. it must provide a detailed account, similar to the one we present above, of how error-based learning can lead to a large short-term structural priming and yet show an inverse-preference effect.

§ 3.4.1.5. **Accumulation of priming.—** How does structural priming accumulate over a sequence of trials? We saw that Ferreira and Bock (2006) expected (although incorrectly) that less-preferred structure should accumulate more long-term priming than more-preferred structure. Hartsuiker and Westenberg (2000) too summoned a long-term storage mechanism as an explaination for the change in baseline levels of syntactic choice. A mechanism for long-term accumulation of priming is crucial for explaining how structural choice is influenced by not just the previous trial, but an entire sequence of trials.

An error-based learning model has a natural way of accumulating information in its internal representation of the stimuli. As learning progresses, the error-based model iteratively updates this internal representation and combines new information with the stored information. Thus each episode of priming adds to the previous and the model

automatically makes decisions based on all the previous episodes that it has encountered.

Since change in the existing representation of the model depends on the size of error, such an account makes very specific predictions about the effect of previous trials on the current trial. Equation 3.4.8 gives us insight into how an error-based model updates its internal representation in light of the stimuli. Each adjustment will depend upon the mismatch between the internal representation $p_x^{(n-1)}$ and the external input p_x^{test}. Information about this mismatch would then be stored by updating the internal representation $p_x^{(n)}$. For a subsequent trial, the error would depend upon the mismatch between this new internal representation and the new stimuli. Since the internal representation itself depended on the error correction during the previous trial, the new error would indirectly depend on the outcome of the previous trial. Thus priming during each trial becomes causally liked to all the previous trials.

But does this error-correction based causal link between different trials explain the pattern of priming obtained over consecutive trials? What would be the predictions of the error-based model when two consecutive trials have the same syntactic structure? Would the priming be the same as when the trials are different? Or would it be more or less? Again, we turn to the theoretical account developed above for the answers. In particular, we investigate the predictions for two cases: (i) the case where two consecutive trials have the same syntactic structure, and (ii) the case where the trials have alternating structure.

Consider a prime trial of type 'y' followed by another of type 'y'. For the sake of example, we take the hypothetical values $\mathcal{F}(p_x^{(1)}) = 0.7$, $\mathcal{F}(p_y^{(1)}) = 0.3$ and $\lambda = 0.1$. Thus, at the beginning of the first trial, there is a 70 % chance that the model would predict a trial to have the structure 'x'. Using Equation 3.4.8 we can calculate that at the end of this trial,

$$\begin{aligned} \mathcal{F}(p_y^{(2)}) &= 0.30 - 0.1\,(0.30 - 1) \\ &= 0.37 \end{aligned}$$

where we have used $\mathcal{F}(p_y^{test}) = 1$ since we know that the outcome of the trial is selection of structure y (with probability 1). Thus the new value for $\mathcal{F}(p_y)$ is 0.37, a change of 0.07. We use this change in the value of $\mathcal{F}(p_y)$ as a measurement of the learning in the system and hence as a measurement of priming.

First let us consider the first case where the second trial is the same structure as the

first. We can again use Equation 3.4.8 to calculate,

$$\begin{aligned}\mathcal{F}(\mathtt{p}_\mathtt{y}^{(3)}) &= 0.37 - 0.1\,(0.37-1)\\ &= 0.433\end{aligned}$$

an increase of 0.063. Now consider the other case where the first trial has the structure 'x' and is then followed by a trial of type 'y'. In this case the value of $\mathcal{F}(\mathtt{p}_\mathtt{y})$ will change after the first and second trials in the following way:

$$\begin{aligned}\mathcal{F}(\mathtt{p}_\mathtt{y}^{(2)}) &= 0.30 - 0.1\,(0.30-0)\\ &= 0.27\\ \mathcal{F}(\mathtt{p}_\mathtt{y}^{(3)}) &= 0.27 - 0.1\,(0.27-1)\\ &= 0.343\end{aligned}$$

In this second case, the probability of classification as 'y' increases by 0.073, which is greater than that in the first case (0.063). Clearly, this account predicts that the amount of change to the internal representation is larger during the second trial when consecutive trials have different syntactic structure. This result makes intuitive sense too. The first trial changes the expectations of the error-based model. If the second trial has the same syntax as the first, the model's expectations are more in line with the stimuli and the amount of learning is small. But if the second trial has the alternative syntax, the model is surprised. This surprise leads to larger error correction and larger priming. Thus an error-based model predicts that different structures during consecutive trials will lead to a larger change in priming as compared to the same structure.

How do these predictions compare with data? Currently, there does not seem to be any study that measures how priming changes during an experiment, though studies exist that show the build up of priming as a result of sequence of trials. Hartsuiker and Westenberg (2000), for example, reported that for the written condition the effect of priming was much stronger when both the previous and current primes were of the same type than when the primes were of a different type. When the primes were of the same type, the difference in priming was around 15% for auxiliary-final responses and around 17% for participle-final responses. In contrast, when the two primes were of a different type the priming reduced to 6% and 5% respectively.

These results are the *opposite* of what is expected from the error-based model, which predicts larger priming when the two primes are different. The above analysis also shows that the priming difference between the two competing structures is

lower when the consecutive primes are of the same type. We can measure the priming difference as $|\mathcal{F}(\mathtt{p}_\mathtt{x}^{(3)}) - \mathcal{F}(\mathtt{p}_\mathtt{y}^{(3)})|$. In the above example, this priming difference is 0.134 when the two primes are the same and 0.314 when the two primes are different. Thus, irrespective of whether we measure priming as the change in $\mathcal{F}(\mathtt{p}_\mathtt{x})$ or as the priming difference, the error-based model predicts larger priming when the two primes were different than when they were the same.[3] Interestingly, the two trials, tested for the repetition of prime by Hartsuiker and Westenberg (2000), were separated by eight items. Therefore, the effect of one priming trial on the other seems to be a long-term effect and the error-based model cannot appeal to short-term priming effects to explain this data.

§ 3.4.1.6. **Amount of priming.—** Another interesting prediction of the above theory is regarding the quantity of priming for stimuli of differing relative frequency. Equation 3.4.6 tells us that the average amount of error depends both on the frequency of the test stimuli and the difference in baseline frequencies of the competing structures, $(\mathcal{F}(\mathtt{p}_\mathtt{x}^{\mathtt{train}}) - \mathcal{F}(\mathtt{p}_\mathtt{y}^{\mathtt{train}}))$. However, if the experiment consists of equal number of both primes ($\mathtt{p}_\mathtt{x}^{\mathtt{test}} = 0.5$), the probability of a prediction error becomes 0.5, a constant. Thus the average amount of priming over the experiment does not depend upon the difference in the frequencies of the competing structures at all. The reason for this is clear from Figure 3.4.4 which shows that each plot passes through the point $(0.5, 0.5)$, irrespective of the difference in frequencies.

How does this prediction match up with experimental observation? Most psycholinguistic experiments consist of an equal number of alternative kinds of primes. Therefore, according to the error-based account, a variation in the frequency of competing structures should be independent of the amount of priming. There has been no experiment that varies the frequency of the structural alternatives and notes the effect on priming. However, it has been repeatedly observed that transitives consistently show lesser priming compared to datives (Bock, 1986; Bock & Loebell, 1990; Bock & Griffin, 2000). One of the differences between transitives and datives is the relative frequencies of their alternative forms. While actives and passives have a large difference in their frequencies, prepositional and double-object datives seem to be better balanced

[3] Although it must be mentioned that this change in priming difference changes behaviour based on the frequency of the primes. When the two consecutive trials use the more-preferred prime, x, the priming difference will increase as a result of consecutive trials using the same structure relative to trials using different structure. Nevertheless, Hartsuiker and Westenberg (2000) reported that the increase in priming for the same structure held for both the more-preferred participle-final and less-preferred auxiliary-final structures.

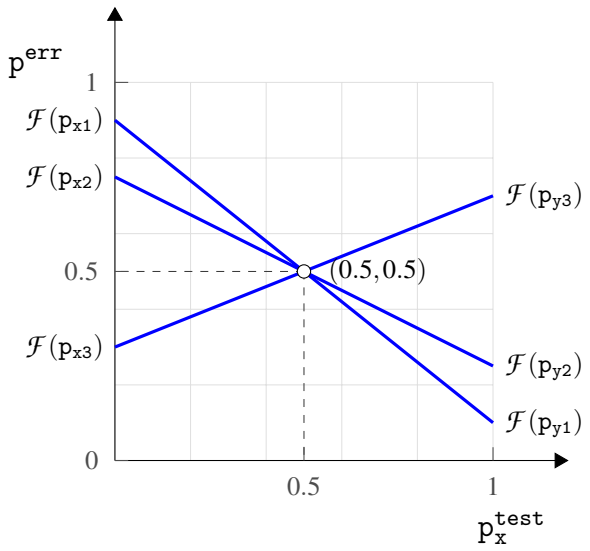


Figure 3.4.4: [Average priming] Three different structural alternatives, $x1-y1$, $x2-y2$, $x3-y3$ are plotted. These three structures have different relative frequencies and therefore, different slopes. All three intersect at $(0.5, 0.5)$

(Bock & Griffin, 2000). If this difference in relative frequencies is responsible for the difference in priming between transitives and datives, then this observation runs contrary to the predictions of an error-based model. However, as Bock and Griffin (2000) pointed out, the difference in priming between transitives and datives could be due to a number of differences in the two structures, including the number of arguments that they express and their likelihood of occurrence.

3.5 Trailing-activation account

We saw at the start of the chapter that learning in model can be supervised or unsupervised. We have been looking at the error-based learning account of structural priming until now. This account uses a supervised learning mechanism that uses the linguistic signal as both the input and as the teacher. However, learning does not necessarily need to rely on a teacher and it can be unsupervised. In this section we look at a conceptual account of structural priming that relies on unsupervised learning. This is the trailing-activation account.

Pickering and Branigan (1998) proposed the trailing-activation account to explain structural priming and its lexical enhancement. Their account is an extension of the spreading activation theory of lemma retrieval proposed by Roelofs (1992), which was

later expanded in Levelt et al. (1999) and is based on spreading-activation theories of phonological encoding (Dell, 1986), semantic processing (Collins & Loftus, 1975) and memory retrieval (J. R. Anderson, 1983). The notion of spreading activation goes further back, developed by Quillian (1962, 1967) to implement human semantic processing in a computer. The difference in trailing-activation and spreading-activation accounts is that of emphasis; while the spreading-activation accounts try to understand the process of information retrieval, the trailing-activation account aims to investigate the effect of time on stored information. Aside from this difference in emphasis both accounts share the underlying theory of information representation and processing. Therefore, to understand the trailing-activation account, it is crucial to understand its precursor – the spreading-activation theory.

§ 3.5.1 The theory of spreading activation

Spreading activation forms the foundation of trailing activation and provides insight into three fundamental concepts of trailing activation: nodes, activation and network organisation. These concepts were formulated to understand the process of information retrieval and to understand them we need to look at the context in which they were developed.

Information retrieval in humans depends on relational properties of memories. While von Neumann machines retrieve information at a pre-specified location instantaneously, humans retrieve information based on memory cues. These cues usually contain incomplete information about a word, a concept, an episode, etc. To complete the information present in the cue, memory needs to be searched. An exhaustive search of memory would provide the correct information, but it would be time-consuming and wasteful of cognitive resources. Fortunately, an exhaustive search is not required as memory is not a random collection of knowledge, but an organised network of information. Words, concepts and episodes are linked to each other and to their features through a well-defined set of relationships. The retrieval mechanism can use the contents of the cue to move through this network, instead of randomly searching the entire memory.

The idea of information retrieval by searching through a network of related concepts is formalised in the spreading-activation theory (Collins & Loftus, 1975). This theory represents each concept as a *node* which is linked to other concepts via its properties. Together, the nodes and their relational links form a network of organised

information – i.e. the system's[4] memory. Given such an organisation of information, the process of search becomes more straightforward. The search begins with a set of cues that index particular nodes in this network. The system can now move, in parallel, along all the relational links until it finds information that meets some criteria. This process of moving along the relational links of the network is the essence of spreading-activation account.

There is no mystery in the use of the term *spreading* which suggests moving, in parallel, along the relational links in search of information. But we have not yet specified who is doing the moving along these links. That is where the term *activation* comes in. So far, the description of the spreading-activation theory assumes a homunculus that traces the paths from one node to a related node in search of information.[5] The concept of activation helps the theory to get rid of this notion of homunculus and replace it with a property of the system. This activation can be defined as a scalar variable associated with each node and each link. The value of this variable indicates the suitability of the node to the search goal.

Now that the abstract notion of a moving through the network has been replaced by a formal variable, we can constrain how this variable changes as the search progresses. Collins and Loftus (1975), for example, constrained that activation must decrease as it spread out along a network path, that it needed to reach a certain threshold for the search to be successful, that it should decrease with time and that the summation of activation over the entire network could be controlled. These constraints allowed them to perform certain predictions about semantic retrieval in memory which could then be compared with experimental evidence. Different constraints about how the activation variable changes with time and with input allow us to explain different experimental findings. We dedicate a large part of the next chapter to investigating how structural priming can be explained by the dynamical properties of this activation variable. A major contribution of the spreading-activation theory is this formal idea of an activation variable and is inherited by the trailing-activation account of structural priming.

Another key conceptual contribution of the spreading-activation theory is the idea of network connectivity and organisation of information. Indeed it is this idea that allowed us to move from an exhaustive search through the entire system to a restricted (and more efficient) search amongst concepts related to the cue. Each con-

[4]Here we use the generic term 'system' to stand for either a computer trying to implement human memory or a cognitive system.

[5]In fact, it assumes a group of homunculi, as information needs to be searched in parallel.

nection in such a network provides a passage for the flow of information. A fully connected network would mean an unrestricted flow of information and a homogeneously connected network would mean that all links carry the same amount of information. The spreading-activation theory treats the strength and nature of connectivity between nodes as parameters of the system. Collins and Loftus (1975), for example, organised the semantic network based upon semantic similarity. This organisation helped them to explain the role of semantic associations during semantic search. We would see below how both the lemma retrieval model, developed by Roelofs (1992), and the models developed in the next chapter use this concept of network organisation to restrict the flow of information.

Even though spreading-activation theory has been developed to explain (and implement) information-retrieval, it provides a general framework for studying the flow and dynamics of information in the cognitive system. We will see below that the concepts of nodes, activation and network organisation are fundamental for understanding how trailing activation can explain structural priming.

§ 3.5.2 The Roelofs (1992) model of lemma retrieval

Roelofs (1992, 1993) and Levelt et al. (1999) applied the spreading-activation theory to lexical and phonological retrieval during language production. We saw in the last chapter that language production involves different stages of processing, each contributing an information structure. A key process during production requires the cognitive system to retrieve a word from the mental lexicon, given a particular concept. It is this process that is modelled by Roelofs (1992) using the spreading-activation theory.

Since spreading activation is an abstract theory, Roelofs (1992) adapted this theory to model language production by specifying the three fundamental concepts of spreading activation: nodes, network connectivity and activation. Let us look at nodes and networks first. These concepts are domain specific – i.e. different domains of memory have different primitives that can be represented by nodes and, of course, they are connected in different ways. When the memory is searched for words, as in the case considered by Roelofs (1992), the network needs to search a network that represents associations of words. It turns out that words are involved in three mutually-exclusive kinds of associations, each of which plays a role in lexical retrieval (Levelt et al., 1999). First, the concept of each word is associated with other concepts: a dog is an animal, it barks, it is closely related to the wolf species, etc. This information is represented at a

conceptual stratum. Next, each concept corresponds to a lemma. Different languages will usually represent the same concept as different lemmas. The concept dog, for example, is represented in different languages by different words: in English it is ‘dog’, in French ‘chien’ and in Dutch ‘hond’. Each language also associates the lemma with its diacritic features. These diacritic features include features such as the number, tense and aspect for verbs which need to be assigned values during sentence production. The language-dependent manifestation of concepts, along with their diacritic features form the lemma stratum. Finally, each word is associated with phonemes that allow a person to express the word as an auditory signal. These associations form the word-form stratum. Thus Roelofs (1992) and Levelt et al. (1999) have developed a tripartite network for lexical memory (Figure 3.5.1) which they use to implement a spreading-activation theory of lexical retrieval.

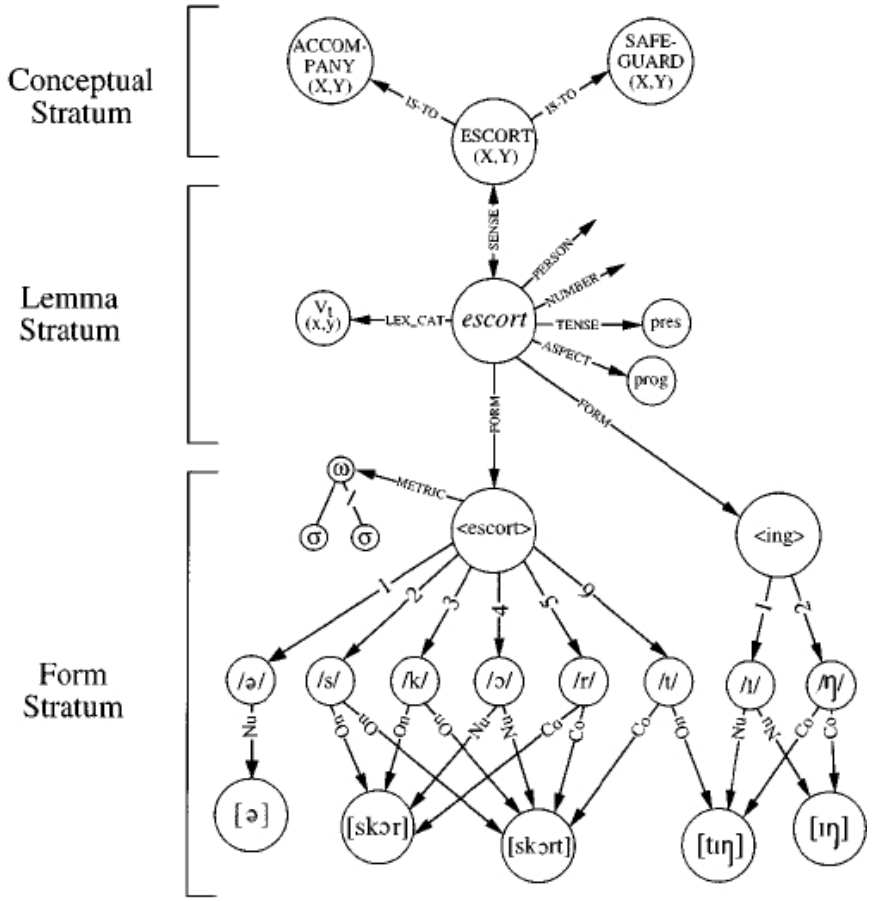


Figure 3.5.1: Fragment of the lexical network underlying lexical access. The network gives an example of the word ‘escort’ in the lexicon. Copied from Levelt et al. (1999).

Given this representation of lexical memory, it only remains to define the rules of

activation spreading to obtain a procedure for lexical retrieval. Roelofs (1992) define activation as a positive real-valued quantity that spreads according to the following equation:

$$a(m, t+\Delta t) \;=\; a(m,t)\cdot(1-d) + \sum_{n\in N} w(n,m)\cdot a(n,t) \tag{3.5.1}$$

where $a(m,t)$ is the activation level of the node m at point in time t, d is the decay rate ($0 < d < 1$), Δt is the duration of a time step and $w(n,m)$ is the weight on the connection from node n to node m. This equation says that the activation of a particular node at any given time is a fraction of its activation at the previous time step and the (weighted) sum of activation coming in from all nodes connected to this node. Thus there are two parts to the equation: decay and integration. As the search progresses, activation moves along the network pathways due to integration, but also decays by a certain factor, $(1-d)$. Once this activation function has been defined, it is simple to understand the process of lexical selection. Usually, the selection is some function of the activation of the nodes. In the simplest case, the node with the largest activation gets selected. In a more complicated case, selection is some function of the activation of the entire network.

The theory takes the notion of nodes, networks and activation from the spreading-activation theory and specifies how these concepts apply to language production. Network connectivity is highly structured and restricted to separate out conceptual, lemma and word-form strata. Nodes are non-decompositional representations of concepts, lemmas, features and word-forms. Activation spreads in a mathematically well-defined manner, specified by equation 3.5.1. Roelofs (1992) use this theory to replicate several experimental findings related to lemma retrieval during language production, including picture naming and picture-word interference task. Levelt et al. (1999) discuss how an extension of this theory can also be used for performing word-form encoding.

§ 3.5.3 The trailing-activation extension

The spreading-activation theory proposed by Collins and Loftus (1975) for semantic search was adapted by Roelofs (1992) for language production. It was originally designed as a generic theory to search memory and when it is applied to production, the theory explains production as a memory search for lemmas and word-forms. Searching results in selection and selection results in production.

Going the other way around, one can say that production depends on the outcome of the memory search. Thus production depends on all the variables that search depends

on. And since search depends on activation, we can say that production depends on activation and its parameters. These parameters are formally arranged in Equation 3.5.1, which shows that activation of a node will depend on the decay rate (d), the connectivity of the network ($w(n,m)$), the activity of all other nodes and the previous activity of the node itself. The decision to produce one word over another will depend on the selection of the node and consequently on these parameters.

Now, the phenomenon of structural priming tells us that the production of a particular structure is related to the choices made about the comprehension or production of the structure in the recent past. Thus production depends not only on the current inputs, but also on past episodes of structural selection. These past episodes themselves must have involved procedures of memory search. Therefore, one can say that if production depends on memory search then evidence from priming tells us that search itself should depend on past episodes of searching. One way to make this happen is to let each episode alter the parameters of Equation 3.5.1, letting the parameters form the causal links between two episodes. In fact, since search proceeds by changing the activation of nodes along a path, it already alters the key system parameter: activation. If activation from one episode of searching can survive till the next episode, then the two episodes will be causally coupled through this residual activation. That is the key idea behind the trailing-activation account of structural priming.

We have started using the term 'structural' priming, but the Roelofs (1992) model was developed for the selection of words. In order to deal with structural selection, Pickering and Branigan (1998) make two observations that allow them to extend the spreading-activation model. The first observation is that structural selection involves choosing how words *combine* with other words in an utterance. An utterance using prepositional dative construction (*The banker gave the gold to the burglar*) combines a noun phrase (NP) with a prepositional phrase (PP) while an utterance using double object dative construction (*The banker gave the burglar the gold*) combines a noun phrase with another noun phrase. If we want to search the memory for structural selection, then the memory can represent different choices as combinatorial nodes (NP-PP or NP-NP). These combinatorial nodes belong to the syntactic level and therefore should be added to the lemma stratum in Figure 3.5.1. The second observation made by Pickering and Branigan (1998) was that repetition of verb leads to an increase in priming. Thus, if priming is due to residual activation in the combinatorial nodes, then this activation is causally related to the activation in the verb nodes. Therefore, Pickering and Branigan (1998) hypothesise links between the combinatorial nodes and

lemma nodes for verbs.

Thus the trailing-activation account extends the spreading-activation theory for lemma retrieval by (a) introducing combinatorial nodes at the lemma stratum, (b) introducing links between the lemma nodes for verbs and the combinatorial nodes and (c) hypothesising that priming is due to residual activation in the nodes from one episode to the other. During production, the memory search will result not only in lemma selection, but also in the selection of a combinatorial node. This latter selection will be governed by residual activation (implementing priming) and input from verb lemmas (implementing lexical boost).

§ 3.5.4 Limitations of the trailing-activation account

Spreading activation was developed as a generic framework for memory search. Roelofs (1992) adapted it for searching the mental lexicon and simulated it for testing lemma selection (Roelofs, 1992, 1993) and word-form encoding (Levelt et al., 1999). While Pickering and Branigan (1998) specify how the spreading-activation theory needs to be extended to account for structural priming, they make a number of assumptions about the operation of their model that require justification or elaboration.

§ 3.5.4.1. **Representations and processes.—** The trailing-activation account is rich in representations but lacking in processes. The representations include lemmas, syntactic constructions and their relationships. Both the lemmas and the syntactic constructions (combinatorial properties) are represented as nodes. The relationships between between these nodes are represented as links. These nodes and links form a network that is accessed during the episodes of language comprehension and production. The only processes specified during either of these episodes is the activation of nodes and links. Beyond this, the account is silent about processes such as how these nodes get activated, how the model learns with repeated activation, how the activation decays, do different traces of activation interfere, etc.

Specifically, the trailing-activation model assumes that both comprehension and production lead to the activation of lemma nodes and the associated combinatorial nodes. This assumption requires that comprehension and production share these representations and that these representations contain traces of memory from each episode. The model does not specify the constraints under which such lemma and combinatorial nodes will be activated and how exactly a node is chosen from its cohort. The spreading-activation theory specified that activation spreads along links and that lemma

selection is made based on the activation (Roelofs, 1992). However, it is not clear whether the same is applicable to the trailing-activation account. And if this is the case, then what is the contribution of the residual activation in the selection of lemma nodes and combinatorial nodes and what proportion of activation is spread out from each node to the connected nodes? The original spreading-activation model also assumes no inhibitory connections between nodes and implements the selection process as a random event that is likely to select highly activated nodes. It is unclear whether selection of the combinatorial nodes in the trailing-activation account takes place through the same process.

The trailing-activation account also assumes that lexical boost is a result of active links between the lemma nodes and combinatorial nodes. Again, the account does not specify how long the link remains active and whether it shows any effect of being repeatedly activated. Pickering and Branigan (1998) assume that the selection of a combinatorial node takes place based on the activation that it receives from the lemma nodes. This mechanism allows them to explain the lexical boost effect. However, it is unclear how other processes can participate in the selection of the combinatorial node, in addition to the selected lemma. For example, if speakers adapt what they speak according to their listeners (Brennan & Clark, 1996), then how does this information play a role in the selection of the syntactic structure. In fact, the trailing-activation account does not explain how this information can play a role in the selection of the lemma node either.

Thus, the major limitations of the trailing-activation account lie in the fact that it is a conceptual model and not a mechanistic one. In fact, some of our questions about the error-based model, such as the relative amount of priming from comprehension to production and from production to production, might be valid for a mechanistic model of the trailing-activation account as well. In the absence of such an account, it is difficult to know what the prediction of such a model will be. Therefore, the trailing-activation account leaves a number of open questions, including:

1. How does activation survive from one episode to the other? Equation 3.5.1 specifies a decay rate (d) for activation. This is the rate at which activation decays within an episode. In the simulations conducted by Levelt et al. (1999), this parameter was set to 0.0240/msec. If activation decays at this rate, then the residual activation will barely have any significance between two trials that are more than 1 sec (1000 ms) apart. An alternative mechanism of decay needs to be specified that can demonstrate how residual activation can account for how struc-

tural priming survives when prime and target trials are separated with variable number of filler trials (for example in Bock and Griffin (2000)).

2. What is the mechanism of learning as a result of an episode of comprehension or production? Pickering and Branigan (1998) only specified that each episode leaves a trace of residual activation. This activation could be as a result of an error-based learning mechanism like the one specified by Chang et al. (2006), or it could be the result of unsupervised learning.

3. What is the mechanism for structural selection? Roelofs (1992) specified a selection mechanism based on the activation of nodes. However, this mechanism is tailored for the experiments that they are studying: picture categorisation, picture-word interference, etc. The structural priming experiments would require its own criterion for selection amongst combinatorial nodes.

4. How does priming accumulate over several trials? We saw that data from Hartsuiker and Westenberg (2000) and Kaschak and Borreggine (2008) showed that structural priming accumulates over a series of trials. How does activation allow accumulation of priming?

5. How are nodes added to the spreading-activation network? This is a question that can be asked of not just the trailing-activation account but also of its predecessors. These accounts assume a long-term memory representation consisting of a network of nodes. Long-term learning should be able to add nodes to the network and modify strength of existing connections. How does this process occur?

These questions indicate that the trailing-activation account specified by Pickering and Branigan (1998) is only a conceptual framework for studying priming through spreading-activation networks. However, only a fully implemented formal model can make predictions that can be tested against experimental observations. The following chapters will present such models and show that they provide a compelling alternative to error-based models.

4 Dynamical systems approach to priming

4.1 Introduction

In previous chapters, we have motivated the need for a theoretical account that can investigate the learning principles behind structural priming. We also saw that one such account exists but fails to explain certain patterns of structural priming and especially the phenomenon of lexical boost. In this chapter we will develop three computational models based on a novel theoretical account of structural priming. These models explore the learning mechanisms behind structural priming and also explain the lexical enhancement of this priming.

Our intention is to model the effect of linguistic processing on the cognitive system. We would like to develop a mathematical description of how the cognitive system learns as a result of linguistic processing. We would also like to investigate what happens to the cognitive system after it has undergone learning. In Chapter 2 we motivated the study of the temporal properties of structural priming. If we are to understand how structural priming changes with time, we must understand how the cognitive system changes after it has undergone learning. We can formally study change in any physical system through the theory of dynamical systems. This theory introduces the mathematical apparatus required to track the evolution of a physical system through time. By applying this theory to the relevant aspects of the cognitive system, we can track the effect of an episode of linguistic processing on the cognitive system. Therefore, we will begin this chapter by introducing the theory of dynamical systems and see how this theory can be applied to the study of structural priming.

Once we have presented this theory, we will build a set of mathematical models that exploit this theory and help us to understand the computational processes that could

underlie structural priming. Specifically, we will present three models, each of which focuses on a particular property of structural priming. The reader might wonder why we present three models and not just one, all-inclusive model. The reason lies in the nature of our investigation. The goal of this chapter (and the thesis) is to explore possible computational mechanisms that could underlie structural priming. In this thesis, we choose a certain set of computational mechanisms that capture the properties of structural priming. However, it is not necessary that these are the only computational mechanisms that could underlie the behavioural phenomena. In future, other studies could exploit a different set of computational mechanisms that can also capture these phenomena. Our goal is to (a) demonstrate the utility of dynamical systems for investigating structural priming, and (b) to gain insight into cognitive processes that might affect structural priming. Each of the three computational models presented in this chapter helps us to gain insight into a cognitive process that could play a key role in amount of structural priming.

While the three models explore different cognitive processes that could affect the amount of structural priming, they also share some core computational mechanisms. Each model assumes that lexical and syntactic information is represented by two separate dynamical systems which participate in both language comprehension and production. During an episode of linguistic processing, these dynamical systems may undergo a change in state. We will explore the nature of this change when we discuss each model. As a result of this change in state, these dynamical systems store the information related to the episode. Furthermore, the behaviour of the dynamical systems in subsequent episodes depends on their state. Therefore, a change in state leaves a trace of linguistic processing which interferes with future episodes. We will see that this trace of linguistic processing eventually leads to structural priming.

Beyond this common implementation of information storage during linguistic processing, the three models diverge in their architecture and mechanisms. Before we delve into the details of each computational model, here is a summary of the key ideas of each model and the results of the simulations:

- MODEL ① explores the nature of connection between the dynamical systems responsible for structural and lexical processing. Previous accounts of lexical boost, such as the trailing-activation account (Pickering & Branigan, 1998) assume that lexical boost is due to a flow of activity along the links between lexical and combinatorial nodes. This model presents an alternative, by assuming no explicit connections between the two dynamical systems. Instead, it proposes that

the two representations are connected due to a global property of the cognitive system, such as its *automaticity* – i.e. how automatically linguistic constructs are selected during production. By performing simulations on this model, we show that such a representational scheme can capture both structural priming and lexical boost.

- MODEL ②, just like Model ①, assumes dynamical systems for structural and lexical processing. However, unlike the first model, this model has explicit links connecting lexical and combinatorial nodes. Simulations conducted on this model show that this architecture can also capture the patterns of structural priming and lexical boost. In this model, we also explore the mechanisms through which structural priming and lexical boost decay. By hypothesising that different types of dynamical systems are responsible for priming and the lexical enhancement of this priming, simulations on this model show how structural priming can persist over a long period of time, while lexical boost decays quickly.

- MODEL ③ extends the second one by building a learning mechanism that is able to retain not just one prime episode, but a sequence of episodes – i.e. this model builds in the mechanism of incremental learning. While the first and second models are quite independent, with different architectures, the second and third models are more intimately connected, with the third model subsuming the second. Simulations conducted on this model show how both structural priming and the lexical enhancement of this priming accumulate over a series of trials. These simulations also reveal the nature of learning and decay in memory that can lead to the patterns of structural priming and lexical boost observed in psychological experiments. One interesting insight from these simulations is that the lexical influence on combinatorial nodes could last over several trials, but might not be detected by some existing psychological experiments used to measure this effect. Further simulations conducted on this model uncover the reason why such a lexical influence could be hidden from experimental observation.

After presenting the theory of dynamical systems (and how it can be used to study structural priming), we present each of these models in succession. The presentation of each model is divided into three parts – the architecture, the formal description of the model, and the simulations conducted on the model. At the end of the chapter, we discuss what these simulations reveal about the cognitive processes that govern structural priming.

4.2 Theoretical Background

Priming is caused by the flow of information from one point in time to another. An episode of linguistic processing at one point in time changes the system in a particular way. After the episode is over, the system still continues to change. These changes, in turn, influence how the system processes information during subsequent episodes. Clearly, if one wants to understand how one episode of linguistic processing influences a subsequent episode, one has to understand how the system changes between the two episodes. These ideas of change and flow of information from one point in time to another are formalised in the theory of dynamical systems.

In this section we develop the theoretical background required to study dynamical systems. We review the relevant literature with a view of constructing a formal model of syntactic choice and syntactic priming. Our treatment of dynamical systems will focus around the properties of these systems that allow us to capture the behavioural phenomena crucial to the study of structural priming. We also describe two elementary units – a node and a network – that the formal models will build upon. In addition, we review literature pertaining to the representation of associations in such networks of nodes.

§ 4.2.1 Dynamical Systems

Following Hoppensteadt and Izhikevich (1997) we use the term dynamical systems to refer to a system of differential equations of the form:

$$\frac{dX}{dt} = F(X), \qquad X = (X_1, \ldots, X_m)^T \in \mathbf{R}^m \tag{4.2.1}$$

The above equation states that the rate of change of variables $X = (X_1, \ldots, X_m)^T \in \mathbf{R}^m$ is some function $F = (F_1, \ldots, F_m)^T$ on those variables. If we know a set of initial conditions X_0, this equation can determine the state of the system at any subsequent point in time. This definition, therefore, formalises the idea of change in the system and demands that we identify a set of variables over which this change is defined. It also requires us to define the set of functions F that will determine the way in which the system will change.

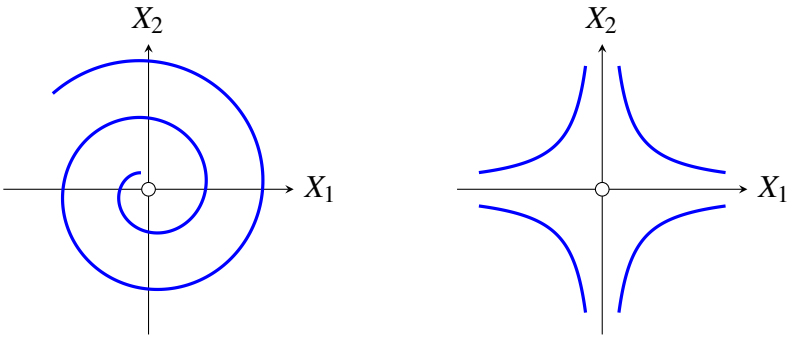
Equation 4.2.1 defines a very general system. It can be used to describe any physical system that undergoes change. The physical laws of the change are encoded in the set of functions F and the properties that undergo change are encoded as variables

X. Thus F stands for the laws of nature. For mechanics these might be the laws of motion, for optics these might be the laws of electromagnetism. Our interest lies in cognition and as we will see, F will encode the laws of cognitive change. Specifically, we are interested in how syntactic structure is chosen and F will encode a mechanism for making this choice.

Once a particular law of nature, F, has been chosen, equation 4.2.1 describes how the properties of the system will change. The solution of this differential equation describes a function $\Phi_t : \mathbf{R}^m \rightarrow \mathbf{R}^m$ which traces the evolution of the variables over time.

$$X(t) = \Phi_t(X_0) \tag{4.2.2}$$

If we know the initial conditions of the system – i.e. if we know the values of various properties of the system at some given point of time – Φ_t will tell us how the system will evolve through time. Thus Φ_t helps us define a *trajectory* of the system. Figure 4.2.1 shows two different trajectories. The figure is drawn for a two dimensional system with variables X_1 and X_2. Each variable is along an axis and the trajectory plots the value of the two variables at different points in time. The two axis only track the values of the two variables and not the value of time. We say that the variables are plotted in the *phase space* and the plane is called the phase plane.



(a) Spiral (b) Saddle

Figure 4.2.1: [Trajectories] Two contrasting trajectories are plotted in the phase plane. (a) shows a stable spiral, while (b) shows an unstable saddle.

The trajectories tell us about the *stability* of dynamical systems. The trajectory of a stable solution converges to a single point while that of an unstable solution diverges. Stability is crucial because, as we will see below, it corresponds to memories in a cognitive system. When the system memorises an event, for example a syntactic choice, it converges to a stable solution. Loss of stability leads to loss of the memory.

Whether or not memories in the human brain really correspond to stable solutions of dynamical systems is a question we will not be exploring in this chapter. This question is part of a wider debate in cognitive science. However, dynamical systems provide a way to model memory and its evolution. They help us formalise the notion of change in memory and in this chapter we will illustrate how we can apply this approach to investigating the effect of priming in language production.

§ 4.2.1.1. **Bifurcations.—** An episode of language processing changes the behaviour of the system. The system acts one way before the episode and another way after. Dynamical system capture such changes in the *qualitative* behaviour of the system through bifurcation phenomena (Hoppensteadt & Izhikevich, 1997). Informally, a bifurcation can be seen as a point around which the system can evolve in (at least) two different ways. Here, the change in the qualitative behaviour of the system is distinguished from a change in its *quantitative* behaviour. As an example, consider the phenomenon of swaying of a trailer at the back of a vehicle moving along a straight path with increasing speed (Seydel, 1999). In the absence of motion, the trailer will lie in a stable state. As the vehicle starts moving, there will be a quantitative change in the state of the trailer. However, beyond a certain velocity, the trailer will suddenly start to sway. Thus there is a particular value of velocity beyond which the system shows a *qualitative* change in its behaviour. This velocity is a bifurcation point for the system.

In the definition of a dynamical system stated in the previous section, the concept of change has been formalised as the change in the values of a set of variables. These variables can be divided into two categories: the internal variables of the system and a set of system parameters. These system parameters are the open variables which can have a range of values. We are interested in those values that can lead to the system showing bifurcations. If a parameter has a value around which the system shows two different dynamics, based on the value of the parameter, then we call such variables *bifurcation parameters*. The value of the parameter at such a point is called the *bifurcation value.* In the trailer example, the velocity of the vehicle is the bifurcation parameter and the velocity at which the trailer starts to sway is the bifurcation value. Formally, a dynamical system $\frac{dX}{dt} = F(X, \rho)$ (where we have separated the internal variables of the system X from the bifurcation parameter ρ) is at a bifurcation point $\rho = \rho_b$ if any neighbourhood of ρ_b contains some ρ_1 such that the qualitative behaviour of the system is different for ρ_b and ρ_1 (Hoppensteadt & Izhikevich, 1997).

The bifurcation characteristics of a dynamical system are interesting to us because

they can alter the system's equilibrium states. Around a bifurcation point the system switches from one kind of equilibrium to another. Equilibria govern the qualitative behaviour of the system and a change in the nature of a system's equilibria is a change in its qualitative behaviour. We can illustrate such a change in a system's qualitative behaviour by studying a particular form of bifurcation called the *saddle-node* bifurcation.

Let us consider a dynamical system for a neuron whose activity depends on the external input to the neuron and some feedback. We formalise the change in activity in the neuron by encoding it as a dependent variable of the differential equation. We also formalise the external input and the feedback by encoding them as independent parameters in the equation:

$$\frac{dx}{dt} = -x + S(\rho + cx) \tag{4.2.3}$$

Here $x \in \mathbf{R}$ is the activity of the neuron, $\rho \in \mathbf{R}$ is the external input to the neuron, and $c \in \mathbf{R}$ is the feedback parameter. The function S is a sigmoidal function which provides a threshold mechanism. Since the change in the activity of the neuron depends on the sum of external input and the feedback, this dynamical system is called the *additive model* (Hoppensteadt & Izhikevich, 1997).

Such an additive model can show different states of equilibrium based on the value of the parameter ρ. When $\rho \to -\infty$, the neuron's activity $x \to 0$. On the other hand when $\rho \to +\infty$, the neuron's activity approaches the equilibrium value one ($x \to 1$). More interestingly, for intermediate values the system switches from this characteristic of *monostability* to *bistability* – i.e. for these intermediate values of ρ the system has two stable states. In fact, the system has three points of equilibrium; two of these are stable (nodes) and one is unstable (saddle-point). Hence the name saddle-node bifurcation. We see that the system shows a *change in its qualitative behaviour* for particular values of ρ. This change in qualitative behaviour, in this case, is the change from monostability to bistability. And since this change in behaviour depends on ρ, this parameter is a bifurcation parameter of the additive model.

§ 4.2.1.2. **Hysteresis.—** Our goal is to study the persistence of priming through dynamical systems. Therefore, we must first define what it means to be primed in terms of such a system. Or at an even more basic level, what do we mean when we talk of an episode of language processing? Each *event* of language processing leads to flow of information through the cognitive system. This flow of information leads to a change in the behaviour of the system. We saw above that a dynamical system

shows a change in its qualitative behaviour when it passes through a bifurcation. Thus, if we want to model the cognitive system as a dynamical system, we have to relate the change in cognition to a change in the qualitative behaviour of the dynamical system. In other words, episodes of language processing can be considered as events that change the qualitative behaviour of the system. Thus, these events can be associated with bifurcation parameters for the dynamical system. When the system encounters an event, its qualitative behaviour changes. The dynamics of the system show how the system is going to evolve and by studying the dynamics we can trace the causal effect of an event through time. This allows us to study the longevity of an event and specifically, since we are concentrating on priming, the persistence of priming.

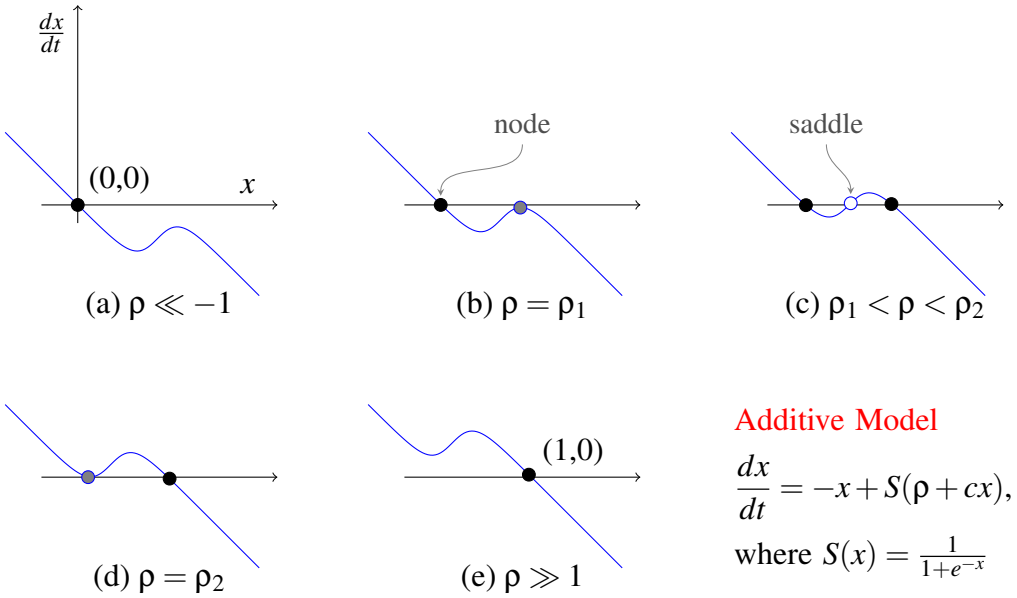


Figure 4.2.2: [Bifurcations] Additive neuron dynamics could have one, two or three equilibria for various values of ρ (Hoppensteadt & Izhikevich, 1997). This figure shows $\frac{dx}{dt}$ plotted against x for various values of the parameter ρ. The system is at an equilibrium whenever $\frac{dx}{dt} = 0$, i.e. whenever the curve crosses x axis. The equilibrium can be a stable node (shown as a filled circle) or an unstable saddle (shown as a shaded circle).

Let us again consider the additive model from the previous section. Figure 4.2.2 shows the dynamics of the additive model for various values of x (the state of the system) and ρ (the bifurcation parameter). Each sub-figure shows how the system evolves for a range of ρ. In the additive model, this bifurcation parameter represents the neuron's external input. Therefore, each of the sub-figures shows the behaviour of the neuron for a range of the external input. It can be seen from the figure that when the

external input ρ is increased beyond a particular value ρ_2 (figure 4.2.2(e)), the neuron will approach the stable state $x \to 1$ (shown as $\bullet$). On the other hand, when the external input is decreased beyond ρ_1 (figure 4.2.2(a)), then the system will approach $x \to 0$. In between these two extremes, when the external input is larger than ρ_1, but smaller than ρ_2, the system exhibits bistability – i.e. there are two stable states at $x = 0$ and $x = 1$. In addition the system also develops an unstable saddle point (shown as $\circ$) in between the two stable nodes. We would like to know which of these equilibriums does the neuron approach when it receives such an intermediate input.

In such cases ($\rho_1 < \rho < \rho_2$) the system's response depends on the previous state of the system (figure 4.2.2(c)). If the system has most recently been pushed above the bifurcation point ρ_2, then the system will approach $x \to 1$. On the other hand, if the system has most recently been pushed below the bifurcation point ρ_1, then it will approach $x \to 0$. Therefore, the system's response depends not only on the value of the input parameter, but also on the history of the system. This phenomenon is known as *hysteresis*, and is illustrated in the Figure 4.2.3. While bifurcations provide us with a mechanism for encoding the impact of an episode of language processing on the dynamical system, hysteresis provides us with a mechanism for encoding how one episode of linguistic processing affects the other. Priming in experimental subjects demonstrates that the linguistic choices during an episode of language processing are not made independently during each episode, but are dependent on episodes in the recent past. Hysteresis provides a mechanism for formalising this dependency between consecutive linguistic choices.

§ 4.2.1.3. **Adaptation.—** So far we have described bifurcations and hysteresis and stated that these phenomena can be used as mechanisms for formalising linguistic processing and the dependency between episodes of processing. We have still not described a mechanism that can be used to formalise how the system changes between episodes. That is, how does the system change when it is left on its own, without any external input (or a constant external input). We know, for example, that human memory shows forgetting and neural firing shows fatigue. In this section, we describe a mechanism that can help us formalise principles of forgetting and fatigue.

The state of a dynamical system depends on the position of its equilibria. As the system evolves, it tries to approach its stable equilibria. An external input to the dynamical system pushes it towards a particular equilibrium. As we saw above, one way of doing this is to push the system beyond a bifurcation point so that the system

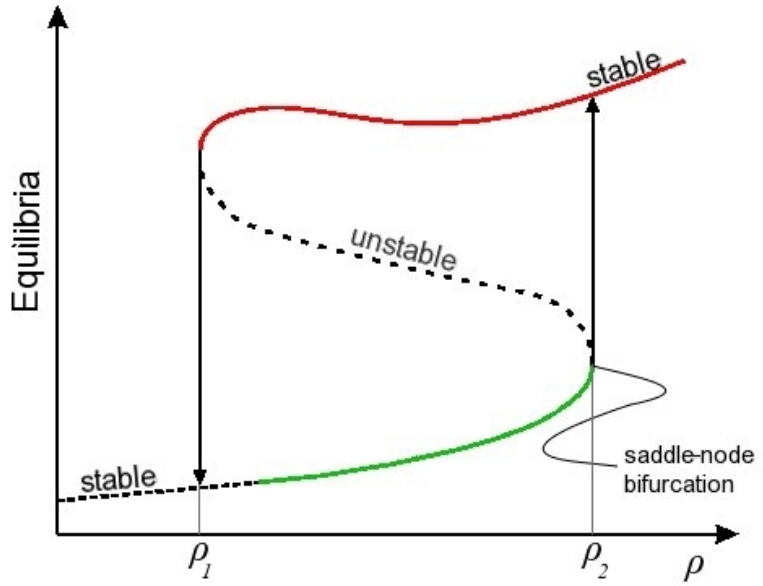


Figure 4.2.3: [Hysteresis] This figure shows the three equilibriums of an additive model against different values of ρ. For a small as well as high values of ρ, only one equilibrium exists. However, for intermediate values the system shows three equilibria – two stable ones and one intervening unstable one. Under these conditions, the system shows hysteresis and settles into the most recent equilibrium. If ρ is swept back and forth across the range $[\rho_1, \rho_2]$, the system will trace out the hysteresis loop shown by the arrows. Adapted from H. R. Wilson (1999).

changes its point of stability. Therefore, events of external input are recorded in the dynamical system through the position of its stability. This is the system's memory.

Forgetting involves destruction of memory and therefore, corresponds to the destruction of the stable equilibria of the dynamical system. This destruction of a system's equilibria is a change in its qualitative behaviour and corresponds to a bifurcation point. Therefore, in order to find a mechanism of forgetting, we need to identify a bifurcation parameter that leads to a loss of stability. Furthermore, this parameter should change its value with the progress of time – i.e. we should be able to describe the change in this parameter with another differential equation. H. R. Wilson (1999) have discussed such a parameter for fatigue in neurons of the visual cortex. Following H. R. Wilson (1999), we call this parameter the *adaptation parameter*, where the term 'adaptation' reflects the observation that when neurons are exposed to a sustained external stimulus, their rate of firing decreases as they adapt to the external stimulus.

Consider the dynamical system:

$$\begin{aligned} \frac{dX}{dt} &= F(X,\rho,A) \\ \frac{dA}{dt} &= G(A,X) \end{aligned} \tag{4.2.4}$$

where $X = (X1, X2)$, $A \in \mathbf{R}$ and F and G are, as usual, vector functions that encode the law for change of variables X and A respectively. This dynamical system is an extension of the one presented at the beginning of the section (equation 4.2.1). We have introduced the parameter A, which, in turn, changes according to its own differential equation. H. R. Wilson (1999) shows that A can act as an adaptation parameter, such that as the system evolves without an external input, A will change such that the system will bifurcate and lose its state of stability. Since this state of stability is the system's memory, this adaptation will, in turn, lead to forgetting.

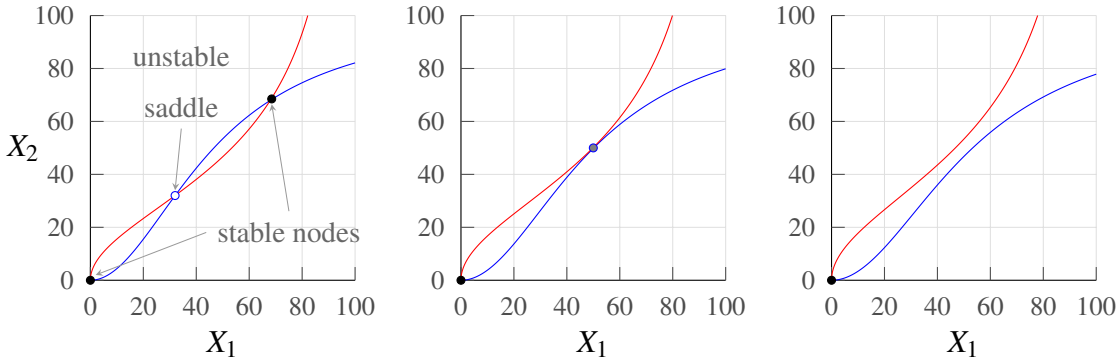


Figure 4.2.4: [Adaptation] Each of the figures shows two isoclines ($\frac{dX_1}{dt} = 0$ and $\frac{dX_2}{dt} = 0$) for a particular value of the adaptation parameter A. The equilibria of the system are given by the intersection of the two isoclines. The three figures are plotted for increasing values of A. For a small value of the adaptation parameter, the system shows three equilibria. As the parameter is increased, two of these equilibria coalesce, leading to a saddle-node bifurcation. As A increases beyond this value, the system shows only one stable equilibrium. Adapted from H. R. Wilson (1999).

This loss of the system's stability can be illustrated by looking at the phase plane of dynamical system for different values of the parameter A. Figure 4.2.4 shows this evolution of the phase plane as the adaptation A increases. Each sub-figure plots the isoclines for the system $\frac{dX}{dt} = F(X,\rho,A)$. The points where the isoclines intersect are the points of equilibrium. In figure 4.2.4(a), the isoclines intersect in three places -

which corresponds to the three values of (X_1, X_2) for which the system is in equilibrium. Two of these points are stable nodes and the third is a saddle point. As A increases, we can see that the saddle point approaches the node till they collide leading to a saddle-node bifurcation. Beyond this bifurcation (figure 4.2.4(c)) the system has only one point of stability – at $(X_1 = 0, X_2 = 0)$. If this system was stable at the non-zero equilibrium node (Figure 4.2.4(a)) before adaptation set in, then this adaptation has led to the loss of the information stored in that state.

§ 4.2.2 Networks and Nodes

So far we have described dynamical systems as a set of differential equations which describe the change in a set of variables (equation 4.2.1). We have also split these variables into bifurcation parameters and internal variables of the system. Finally, we have seen an example that relates the internal variable to the activity of the neuron and the bifurcation parameters to its external input and the feedback (Equation 4.2.3). We would now like to extend this idea and consider larger systems which consist of not just a single internal variable – such as the activity of a node – but a group of such variables, each of which influences the other. Such interactions between a group of variables are best studied using network theory which allows graphical representation of the variables and their relationships as nodes and networks. In this section we will first formally describe what we mean by such a node and then we will consider a small network of two nodes that allows us to implement an interesting behavioural phenomenon.

§ 4.2.2.1. **Node.—** Each node in the model can be seen as a mathematical transformation $\mathbf{T}$, that transforms its input to a scalar output: $x_j = \mathbf{T}(\mathbf{x})$ (see Figure 4.2.5). The input vector $\mathbf{x}$ is the stimulus and the scalar output x_j is the response. In Section 4.2.1.0, we saw how the additive model of Equation 4.2.3 shows saddle-node bifurcation and hysteresis. This additive model sums its inputs and performs a nonlinear transformation on the sum – the sigmoid transformation. We intend to implement each of our nodes as an additive model and for this reason, we need to define this nonlinear transformation. Specifically, we need a transformation that can perform thresholding so that the node does not produce an output response until the it gets an input over a specific limit.

Following H. R. Wilson (1999), we choose the Naka-Rushton function (Naka & Rushton, 1966) to define the nonlinear transformation $S(\cdot)$. This function has the useful

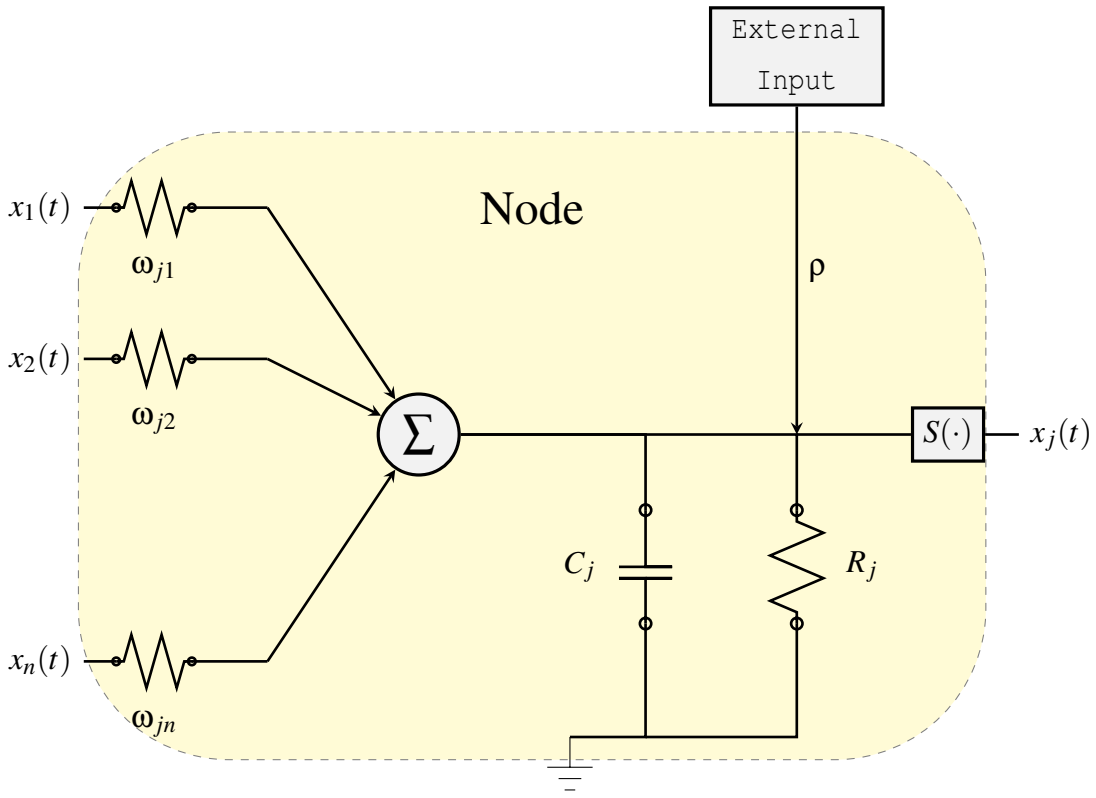


Figure 4.2.5: [Node] Each node is an additive model that sums its inputs (shown as Σ) and performs a nonlinear transformation (shown as $S(\cdot)$). The node also performs temporal integration of the input signal based on the dynamics. In this respect, the node acts as an electrical RC circuit. The time constant of the impulse response function, τ_j, is given by $\tau_j = R_j C_j$, where R_j is the resistance and C_j is the capacitance. Adapted from Haykin (1999).

characteristic of providing a good fit for the cortical neural response to stimuli. By implementing this nonlinear transformation we keep the implementation of a node in our network close to the physiological mechanisms that implement these processes. The function relates the value of an input stimuli x to the response of the node $S(x)$:

$$S(x) = \begin{cases} \dfrac{Mx^N}{\hbar^N + x^N} & for \quad x \geq 0 \\ 0 & for \quad x < 0 \end{cases} \tag{4.2.5}$$

where x is the input to the function, M is the maximum activation, $\hbar$ is semi-saturation constant (the point at which $S(x)$ reaches half its value), and N determines the slope of the function. Figure 4.2.6 plots the function for different values of N. We can see the general sigmoid shape of this function in the figure.

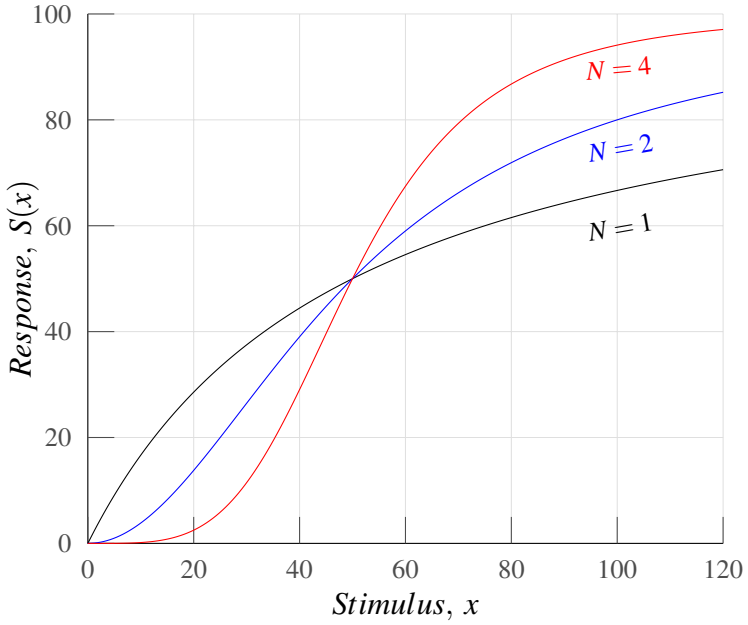


Figure 4.2.6: [Naka-Rushton function] $M = 100$ gives the peak response (y axis) and $\hbar = 50$ gives value of the stimulus (along the x axis) for which response is half of peak response. Adapted from H. R. Wilson (1999).

§ 4.2.2.2. **Two competing nodes.—** We mentioned in Section 4.2.1.1 that an additive model could show hysteresis. This means that if such a system were to implement syntactic choice, then this choice would depend on both the current input, and on the system's past – a highly desirable characteristic since we are studying priming. However, to complete the story, we need a mechanism that not only shows hysteresis, but also implements choice between competing structures. Though the single node in an additive model can show hysteresis, in order to implement competition, we need to look at a network that consists of more than one node.

Consider the following dynamical system in which two nodes are connected to each other via inhibitory links:

$$\frac{dx_1}{dt} = -x_1 + S(\rho_1 - cx_2)$$
$$\frac{dx_2}{dt} = -x_2 + S(\rho_2 - cx_1)$$

Let us look at the dynamics of this system. Each node is an additive model. However, unlike the additive model that we previously considered, here the two nodes have mutually inhibitory connections. Hence the activation of one node forms the (negative) feedback to the other. We want to study how this dynamical system evolves, whether

it reaches equilibrium and what are the points of its equilibrium. This information is given by the system's isoclines. Each isocline traces the curve along which a node is at equilibrium. Figure 4.2.7 shows the isoclines for each of the two nodes. We can see that the isoclines intersect in three different places. Two of these fall along the x axis and y axis respectively. It can be shown (H. R. Wilson, 1999) that each of these equilibriums is stable. The third equilibrium, marked (κ, κ) in the phase plane, is an unstable saddle point. As in section 4.2.1.0, we have a dynamical system with two nodes and a saddle point. At each of the stable equilibrium points, one of the nodes has a positive activation, while the other one has been suppressed and has no activation. The inhibitory connections mean that there is competition between the nodes. One of the nodes wins the competition and completely suppresses the other. Therefore such networks are called **winner-take-all** (WTA) networks.

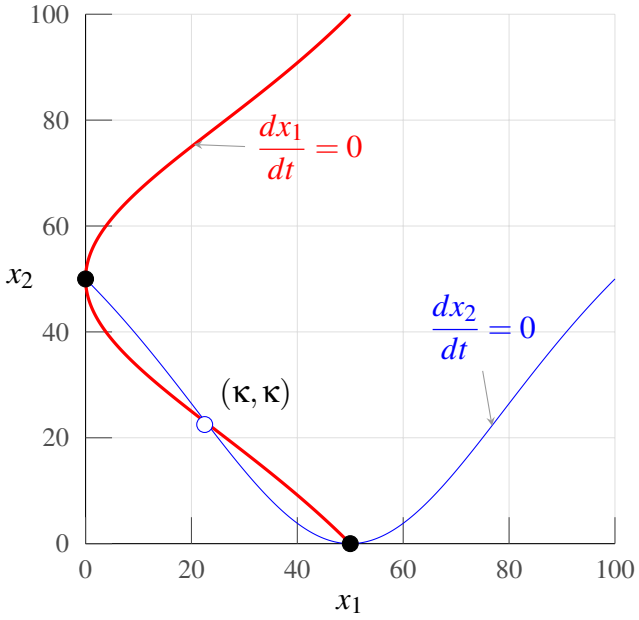


Figure 4.2.7: [Isoclines] Each isocline shows the equilibrium values for the corresponding node. Therefore, the intersection of the two isoclines shows the points of equilibrium for the whole system. This system has two stable nodes (black dots) and one unstable saddle (white dot).

The qualitative behaviour of the dynamical system can be predicted from its bifurcation parameter. The bifurcation parameter for the additive model of Equation 4.2.3 was the external input ρ. For the WTA system with inhibitory connections, this parameter is the difference of the two external inputs $\rho_1 - \rho_2$. If this parameter is below a certain value (say, θ_1), then the first node will win the competition. If the parameter

is above a certain value (say, $\theta_2 > \theta_1$), then the second node will win the competition. For $\theta_1 < \rho_1 - \rho_2 < \theta_2$ the system could be in either of the two stable equilibriums. In this case the equilibrium will depend on the system's past. Thus this WTA network will show hysteresis.

The simple dynamical system with mutually inhibitory links shows two essential properties we were looking for: it implements competition and shows hysteresis. Therefore this system can be used to implement choice in our model. When we describe the network architecture of our models, we will see that nodes in each layer are connected via mutually inhibitory links. Thus each layer can be seen as a dynamical subsystem that shows competition and hysteresis.

§ 4.2.3 Binding

Syntactic choice during sentence production is influenced by two factors:

1. Syntactic choices made in the recent past (structural priming effect).

2. Lexical context of the syntactic choice (lexical boost effect).

Dynamical systems allow us to understand how information changes over time and, consequently account for the structural priming effect. But in order to account for the lexical boost effect, we must specify how lexical and syntactic information is associated. In Section 2.2.3 (page 27) we noted the ongoing debate about the dependence between speakers' lexical and syntactic knowledge. In this section, we will consider how the influence of one form of information on the other can be represented.

Information represented in one part of the system can influence information in another part through two mechanisms. The first mechanism is that of simple association, frequently shown in connectionist models as a link between the two representations. In this mechanism, the two representations are directly connected to each other. Activation of one representation directly influences the activation of the other representation. An alternative mechanism is where the two representations are not directly connected to each other, but through another intermediate representation. In this mechanism, activation of one representation can influence the activation of the other, but only if the intermediate representation is also activated. Thus the flow of information from one representation to the other becomes conditional on the state of a third representation which connects the two.

We need to choose between these two mechanism of information flow and this choice depends on the properties of the relationship that we want to encode. During language production, linguistic constructs are temporarily related with each other in a highly structured manner. These relationships are temporary as they exist only for a short period of time – e.g. phonetic relationships exist for the duration of a word while syntactic relationships exist for the duration of a sentence. Also, each construct is not arbitrarily related to every other construct, rather this relationship is highly structured. For example, if the relationships between syntax and words were arbitrary, then it would be impossible to distinguish the sentence *John gave the book to Mary* from *Mary gave the book to John*. Not only are the words *John*, *Mary*, *give* and *book* associated with each other, but they are associated in a specific structure. We can conclude that the relationships between linguistic representations need to exists only between relevant subsets of linguistic constructs and they need to exist temporarily.

While the mechanism of directly associating representations is simple to implement, it suffers from the disadvantage that the influence between the two representations is simply governed by the frequency of their co-occurrence. The linguistic relationships generated by such an associative mechanism are neither temporary – since they accumulate the long-term frequencies of co-occurrence – nor structured – since they are based on statistical and not structured relationships.

The other mechanism of information flow – through an intermediate representation – overcomes these limitations by having this intermediate representation serve as a gate between the two representations. If this common representation is active, then one representation (say word-form) influences the other (syntactic choice). Otherwise, the two representations stay independent. This intermediate representation can exist for a short period of time and allows us to implement the temporary nature of the relationship. We will see below that this mechanism is advantageous to us because it allows us to study the time course of lexical influence on syntactic choice. It also allows us to represent the fact that not all constructs that co-occur in a sentence will be related to each other. Only those relationships that have an active intermediate representations will be related.

However, in order to implement this mechanism, we need to specify what this intermediate representation could be. In this chapter, we restrict ourselves to the relationship between the word-form of a verb and the syntactic construction that the verb appears in. The intermediate representation needs to store this relationship in a temporary manner. While we will not be concerned with the structured nature of the

relationship (e.g. the verb acts as the predicate and the subject and objects of the sentence act as the arguments to this predicate), we would like to use a representational mechanism that can be generalised to capture this structured nature of the relationship as well. In the next chapter, we will use this generalisation and expand our representational scheme to include structural relationships between different parts of a sentence.

The problem of explicitly storing structured relationships between representations is well studied and is known as the *binding problem* (Fodor & Pylyshyn, 1988; Hummel & Biederman, 1992). The binding problem is not restricted to representation of language, but extends to other modalities. For example, the binding problem needs to be solved in our visual system if it represents an object as the conjunction of its features. The early visual cortex seems to be good at detecting different features of objects, such as orientation, colour and location. This means that if the cognitive system wants to represent an object as a whole, then this information about the orientation, colour, location, etc. needs to be integrated somewhere. Thus we need a mechanism for representing the conjunction of these features.

There are two ways to represent such conjunction of features: (a) through *static* binding, and (b) *dynamic* binding. Static binding uses an explicit symbol, or a node, to represent the conjunction. Thus the conjunction of a colour and an orientation is represented by a node. If there are three possible colours and four possible orientations, then this requires twelve nodes (3×4). This representation has the obvious disadvantage of spatial complexity – i.e. as the number of symbols increase, the nodes required to represent the conjunctions will increase exponentially. The advantage, however, is the simplicity of representation as it only requires a representational unit to encode an association.

Static binding suffers from spatial complexity because all the conjunctions are explicitly represented at the outset. Some of these conjunctions will never be used during the processing of the system. This limitation can be overcome if the system can temporarily bind representations. This kind of conjunction is called *dynamic* binding. Usually, the system uses a particular variable to represent the conjunction of properties. A popular variable for dynamic binding is temporal synchrony (von der Malsburg, 1981). Binding through temporal synchrony assumes that units that code for each representations fire at a particular frequency. Two patterns of activation can then be bound together by synchronizing the firing frequency of units participating in the two patterns. Gray and Singer (1989) have given physiological evidence for the existence of such synchronous firing of neurons. Shastri and Ajjanagadde (1993) have

developed a model that can be used for natural language reasoning and uses temporal synchrony for representing structured relationships in sentences.

In our current study, we are not seeking a complete model of language production. Instead, we want to design a formal model that serves as a proof-of-concept for our premises about priming and lexical boost. Therefore, we adopt the simpler representation of static binding over the more computationally complex dynamic binding. This will make our design more simple and our analysis more lucid. However, the results do not depend on the nature of the binding and the same arguments can be made, *mutatis mutandis*, for a dynamic binding solution as well.

4.3 Model One

As stated in the Introduction to this chapter, we will consider three models for structural priming in this chapter, each based on the theory of dynamical systems. While the second and third model provide a more accurate implementation (and extension) of the trailing-activation account, the first model presents an intriguing digression. This model investigates the range of information that speakers use while making structural decisions. The trailing-activation account assumes that lexical information influences structural decisions through a set of associative links, but this model explores an alternative possibility for such a flow of such information. Models ② and ③ will return to the trailing-activation idea and show how associative links can also account for lexical influence on structural decisions. We present this model first because it is the simplest and helps us introduce some basic dynamical systems that will be used in the following models.

The act of speaking involves making decisions at various levels of processing. Levelt (1989) divides production into three stages: (i) Conceptualisation, during which speakers decide what they want to say, (ii) Formulation, during which speakers choose the linguistic construction and finally (iii) Articulation. Each stage involves a group of processes that make decisions amongst a set of competing alternatives. The speaker can employ a range of considerations while making each of these decisions. At one extreme, each decision can be made *strategically* by considering a broad range of social and inferential premises. Examples of such strategic processes are given by Brennan and Clark (1996) and Brennan and Metzing (2004), who propose that interlocutors in a dialogue explicitly model their audience and carefully "design" their utterances based

on this audience. At the other extreme, each linguistic decision can be made by a set of fast, *automatic* processes, without explicitly considering this broad range of premises. An example of this automatic process is structural priming where interlocutors show a tendency to repeat the construction that they have just heard.

Garrod and Pickering (2007) discuss how the stages of language production show a mixture of automatic and non-automatic processes. Following Bargh (1994), they divide automaticity into four components – *awareness*, *intentionality*, *efficiency* and *controllability*. Garrod and Pickering (2007) argue how each stage of language production can fulfill a varying number of these criteria and therefore shows a graded amount of automaticity. In particular, they discuss how the process of grammatical encoding shows features of both being non-automatic (it is partly open to awareness and competes for central resources) and being automatic (it shows structural priming). Finally, Garrod and Pickering (2007) also discuss evidence suggesting that interlocutors in a dialogue show a variable amount of alignment which suggests that the degree of automaticity in each stage is not fixed, but variable. They suggest that speakers can overcome automatic processes of alignment through effortful processing if they have a strong interest in not aligning with their interlocutor.

A complete model of structural selection needs to show how subjects can employ these automatic and strategic considerations while making structural decisions. This is a non-trivial problem. The model needs to implement a structural decision process in such a way that it shows priming. Secondly, the model needs to show how this priming would increase under lexical repetition. Finally, the model needs to show how priming and lexical boost are affected by the change in the degree of automaticity. In this section, we develop a model that makes structural decisions and allows us to vary the overall degree of automaticity in the system. Before describing the architecture of the model, we review the architecture of an existing model and see how we can change it using the theory of dynamical systems.

§ 4.3.0.1. **A flashback of trailing activation.—** We discussed in Section 3.5 how Pickering and Branigan (1998) extend the lemma retrieval model proposed by Roelofs (1992, 1993) into a conceptual model of structural priming. Pickering and Branigan (1998) found, through a series of experiments, that structural priming was influenced by the repetition of the verb between prime and target trials (the lexical-boost effect). They argued that this result showed that syntactic decisions are influenced by their lexical context. The original spreading-activation model described by Roelofs (1992)

contained only the syntactic *category* information with each lemma. Therefore, this network was insufficient for explaining the lexical-boost effect, prompting Pickering and Branigan (1998) to propose a revision of this model.

In this revised conceptual model, the lemma nodes are connected to a constellation of grammatical nodes which represent grammatical properties such as the tense, aspect, number, etc. and the combinatorial properties of the node. These combinatorial properties correspond to the procedural knowledge (i.e. knowledge of phrase structure rules) associated to a lemma (Pickering & Ferreira, 2008). In this chapter, we are only concerned with the lexical influence on the word order of utterances. Therefore, we will ignore all other links of the lemma nodes except its link with the combinatorial nodes. This isolated set of relationships between two lemma nodes, Give and Send and two combinatorial nodes, NP-PP and NP-NP, is pictorially shown in Figure 4.3.1.

The extension of the model proposed by Pickering and Branigan (1998) emphasises that syntactic information is not stored independently, but in its lexical context. This lexical context is associated with syntactic information through a link. Pickering and Branigan (1998) propose that the lexical context of syntactic structure is encoded by making this link 'active'. An active link allows the flow of activity from one node to the other and therefore, in subsequent trials, syntactic structures receive a *boost* from lexical nodes connected using an active link. In Figure 4.3.1 we pictorially represent the active link between `Send` and `NP-PP` through a thick edge connecting the two nodes.

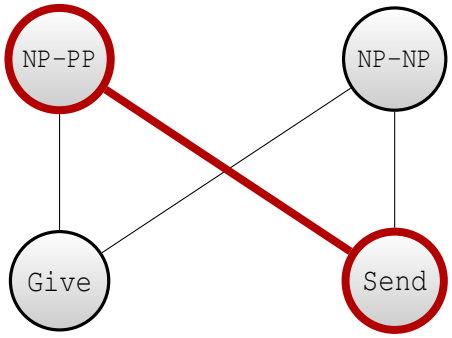


Figure 4.3.1: [Trailing-activation account] Each lemma is associated with a combinatorial node, which governs how the lemma combines with other words in the utterance. Active nodes and links are highlighted.

This revised model, being a conceptual rather than a computational account, leaves many aspects unspecified including the nature of associations between the lexical and combinatorial nodes, the dynamics of the nodes themselves, the mechanism of syntactic choice and the nature of learning with each episode of language production or comprehension. In this section, we will try to investigate the specification of each of these aspects and arrive at an architecture that can explain temporal aspects of syntactic priming and its lexical enhancement.

§ 4.3.1 Architecture

We design our model based on the conceptual model proposed by Pickering and Branigan (1998) (Figure 4.3.1). As the first step, we change this model to include a module that controls the overall degree of automaticity in the system. Then, we assume that this degree of automaticity influences both structural and lexical decisions. While it is true that lexical and syntactic decisions show different degrees of automaticity (Garrod & Pickering, 2007), we assume that these decisions can also be controlled by an overall level of automaticity in the system. This *overall* level of automaticity can, for example, reflect the amount of control that an interlocutor wants to exert over alignment. When a speaker is interested in not aligning with their interlocutor, the system can encode this state by having a low level of automaticity. To reflect the assumption that an overall level of automaticity influences both structural and lexical decisions, we connect this module to lexical and structural layers. Thus, our new system introduces an implicit causal link between lexical and structural layers – through the overall level of automaticity in the system.

Previously, it has always been assumed that lexical boost is caused by an explicit causal connection between lexical and syntactic layers (Pickering & Branigan, 1998; Pickering & Ferreira, 2008). This causal connection has often been represented as associative links between the two layers (Figure 4.3.1). However, we have seen that one can hypothesise an alternative causal connection between lexical and structural layers – through a module that controls the degree of automaticity in the system. In our first model, we test this hypothesis by removing the overt associative links between lexical and structural layers and investigating if the system still shows lexical boost. We call the cognitive process that controls the overall level of automaticity in the system as the system's level of *arousal*. A high value of arousal is related to a low degree of automaticity and vice-versa. In this sense, arousal can be closely compared to the awareness component of automatic processing. The more automatic a process is, the less likely is the speaker to be aware of that process.

Figure 4.3.2 shows a pictorial description of our first model. This figure shows that the model consists of three parts: a lexical layer, a syntactic layer and the Arousal module. In Pickering and Branigan (1998), two nodes in the same layer are not linked to each other. However, in the current model, nodes in the same layer are connected via inhibitory connections. As we will see below, such inhibitory connections lead to competition between nodes and can implement the process of choice between the

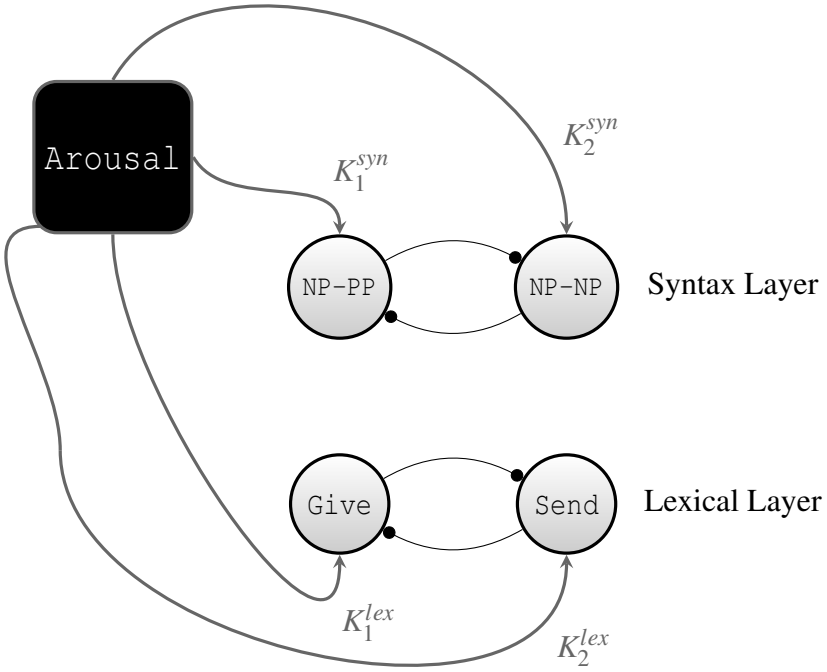


Figure 4.3.2: [Model ①] Each layer shows a representational cohort. The nodes within a layer are connected through inhibitory links. The input of each node (K_i) is connected to a module that maintains the overall level of arousal in the system.

nodes. The third component of the model is an Arousal module. The output of this arousal module is linked to the external input received by the lexical and combinatorial nodes.

§ 4.3.2 Formal Description and Dynamics

In this section we present the precise formal details of the model shown in Figure 4.3.2. We discuss the mathematical details of the connectivity of the model, the dynamics of each node and a systematic method of determining the parameters of the model.

§ 4.3.2.1. **Dynamics.—** Our first model consists of two independent winner-take-all layers. Each node within a layer is connected with an inhibitory link to all other nodes in the layer. For our example, we consider only two nodes in each of the layers. The dynamic equation for the rate of change of activation of each node in $Layer1$ and

*Layer*2 is:

$$\frac{dE_i}{dt} = \frac{1}{\tau}(-E_i + S(K_i - c\sum_{j\neq i} E_j)) \tag{4.3.1}$$

where E_i is the activation of the node, K_i is the external input, c is the feedback parameter, and τ is the time-constant for rate of change of activation of each node (H. R. Wilson, 1999). Therefore, $(K_i - c\sum_j E_j)$ is the net input to each node – the difference between external input and sum of inputs from inhibitory connections. Note that the general form of the equation is the same as the equation for the additive model. The external input ρ has been replaced by K_i. The feedback is negative and depends on the sum of activation of all other nodes[1].

Let us consider the behaviour of this system during a prime trial. The network is symmetric – the strength of the (inhibitory) connections from each node to the other are identical ($c_i = c\ \forall i$). Therefore, if the initial conditions are the same for both the nodes, then the winner is decided by external input. Each external input, K_i, tries to pull the activation of the node towards its own value and each node tries to suppress all other nodes. Since the system always starts from the *rest state* ($E_i = 0$) at the beginning of a prime trial, the winning node at the end of this trial is completely dependent on the external input. The network makes a choice and the winning node is picked by external input.

As discussed above, once the network makes such a choice, it sticks with it – i.e. shows hysteresis. The bifurcation parameter indicates whether the network will show hysteresis or be biased towards one of the nodes. As discussed above, this bifurcation parameter is the difference in external inputs $K_1 - K_2$. Depending on the value of the parameter, one of the nodes will achieve maximum activation – we call this node being in the ON state. The other node will have activation close to zero – we call this node being in the OFF state. If during the priming trial, one of the nodes was turned ON and the other OFF, then the network will have inertia towards keeping each node in its existing state. In order to overcome this inertia, a greater difference of inputs ($\Delta K = |K_1 - K_2|$) is required.

Therefore, during the target trial, the network is driven by two contrasting forces. The first is hysteresis – which provides the network inertia – and the second is external input – which pushes the network towards the node with larger external input. These forces might pull the network towards the same node, or they might be in opposition, with greater external input for the node that has lost the competition.

[1]Model ① has only two nodes, so the summation sign, in this case, is redundant.

§ 4.3.2.2. **Determining external stimuli.—** Since we are studying priming, we are interested in how the system behaves once it has been primed. As we will see below (section 4.3.3), it is fairly straightforward to determine amount of external input during the priming trial. This is because a priming trial is completely governed by the external input. However, during testing, this choice of external input is not so straightforward. At first glance, one might think that the system can be tested by simply providing an equal external input for both the nodes. But doing so would mean that the system is completely determined by its internal state. As we will see, this implies that we will have 100% priming. Subjects do not show 100% structural priming, because each production trial receives a novel set of semantic constraints under which to choose a structure. Thus, making a structural choice finds a balance between relying on the system's memory and fulfilling a set of higher level (semantic or attentional) constraints. This variation in higher level constraints can be modelled in our system with a variation in the external inputs. Specifically, it can be modelled through a variation in the value of the difference in external inputs ΔK.

So we can reformulate the problem of determining external inputs to the problem of determining a value for ΔK_{test}, the difference in external inputs for a typical target trial. Because we want each trial to be independent, we want to assign this ΔK_{test} randomly, with a certain probability distribution function, $p(\Delta K_{test})$. What should this pdf look like? Our first constraint comes from the knowledge that we do not want this external input to be biased towards any of the two combinatorial nodes. This gives us the condition:

$$P(\Delta K_{test} > 0) = 0.5 \tag{4.3.2}$$

where $P(a > b)$ is the probability that a will be greater than b. This equation helps us constrain the shape and the mean for the probability distribution: the mean ΔK_{test}, μ, should be zero and the distribution should be symmetrical about this mean. Several distributions satisfy this criterion. For instance, the Gaussian distribution, $\mathcal{N}(\mu, \sigma)$, with $\mu = 0$, the uniform distribution with $\mu = 0$, the bimodal distributions symmetrical about the y axis all have $P(X > 0) = 0.5$ (figure 4.3.3).

Our second constraint comes from the absolute value of the difference between the external inputs, $|\Delta K_{test}|$. This difference in external inputs governs whether external input will be able to overcome hysteresis. If ΔK_{test} is below a certain value, say $-\theta_1$ (the bifurcation point), then the first node will win the competition, irrespective of the system's past. Similarly, if $\Delta K_{test} > \theta_2$, the system crosses the other bifurcation point and again there is only one stable state: the second node comes ON and suppresses

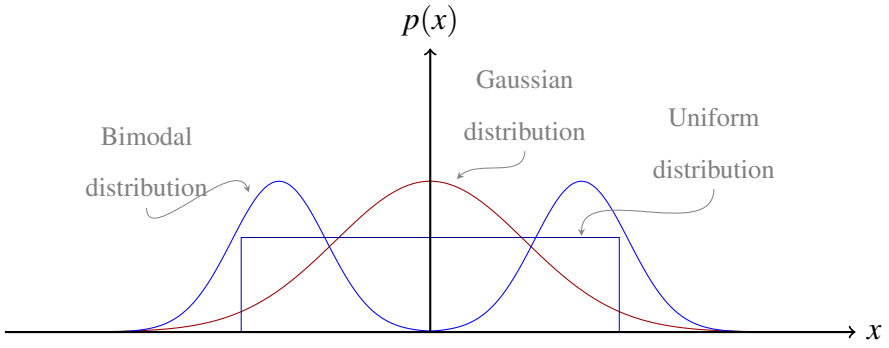


Figure 4.3.3: [Probability distributions] Gaussian, uniform and bimodal distributions, symmetrical about the y axis.

the first node. For a symmetrical system, $|\theta_1| = |\theta_2|$, and we can replace them by a single constant, θ. So, when $|\Delta K_{test}| > \theta$, the external input is the only parameter that determines the system's equilibrium – there is no priming. For intermediate values, $|\Delta K_{test}| < \theta$, the system has two stable states and the choice of the state depends on the system's past. Here, hysteresis dominates external input; the system shows 100% priming.

In a realistic model neither hysteresis, nor external input, will completely dominate the other. This means that the value of $|\Delta K_{test}|$ should be less than θ with a certain probability, say π ($0 < \pi < 1$). Equivalently, we can say that the external input will dominate hysteresis with a probability, $1 - \pi$.

$$P(|\Delta K_{test}| < \theta) = \pi, \qquad 0 < \pi < 1 \tag{4.3.3}$$

Again, this constraint can be fulfilled by several distributions for $p(|\Delta K_{test}|)$. We choose the Gaussian distribution, $\mathcal{N}(\mu, \sigma)$, because it provides us two parameters that will allow us to control the position and the shape of the distribution. Also, the central limit theorem means that the distribution makes fewest assumptions about the distribution of the stimuli.

$$\mathcal{N}(\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-x^2}{2\sigma^2}\right) \tag{4.3.4}$$

The shape of the Gaussian distribution is shown in figure 4.3.3. The distribution is symmetrical about its mean, μ, which governs the location of the distribution. The spread of the distribution around its mean is governed by its second parameter – variance, σ^2. The smaller the value of σ^2, the more concentrated is the distribution about its mean. Larger values of σ^2 will result in larger probability of obtaining values away from the mean. Thus, if $p(|\Delta K_{test}|) = \mathcal{N}(\mu, \sigma)$, then the most likely value of $|\Delta K_{test}|$

(its *expectation*) will be μ and the spread of $|\Delta K_{test}|$ values about the mean will be governed by σ^2.

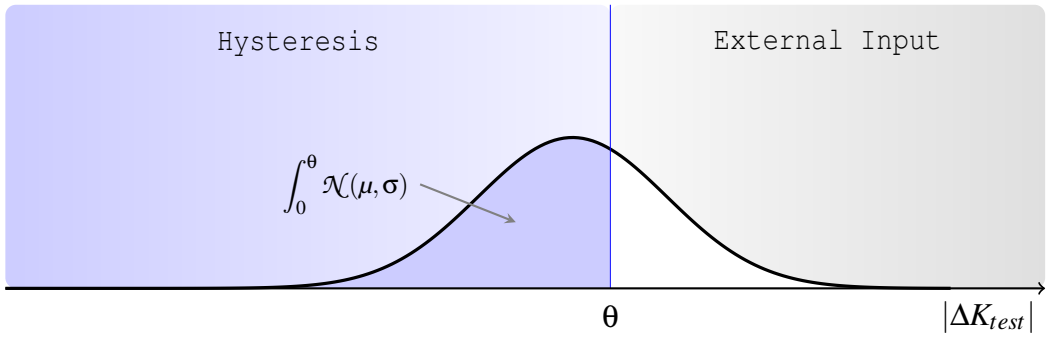


Figure 4.3.4: [Hysteresis vs External input] The value of $|\Delta K_{test}|$ determines whether hysteresis or external input will dominate. If $|\Delta K_{test}|$ is picked from a Gaussian distribution, then probability of hysteresis dominating external input is shown as the shaded region under the curve.

Now, we can restate our problem of determining the external stimulus as determining the parameters μ and σ for a Gaussian distribution, given a dynamical system with the boundary θ and a mean probability of hysteresis, π. From equation 4.3.3 and our assumption of $p(|\Delta K_{test}|) = \mathcal{N}(\mu,\sigma)$, we can deduce

$$\int_0^\theta \mathcal{N}(\mu,\sigma) = \pi \tag{4.3.5}$$

This equation gives us the constraints using which we can determine the mean and the variance of $p(|\Delta K_{test}|)$ in terms of the probability that the model will show hysteresis, π. These constraints are pictorially represented in the Figure 4.3.4. The shaded area under the curve is the probability that $|\Delta K_{test}|$ is less than θ. By moving μ, we will be moving the distribution along the x axis, while the boundary between the forces of hysteresis and external input remains fixed at $|\Delta K_{test}| = \theta$. If we align μ with θ, then each force will dominate the other exactly 50% of the time. Furthermore, we can use well known facts about the Gaussian distribution, such as the probability of a variable assuming a value within standard deviation σ of the mean is 0.68 [2], to calculate appropriate values of μ and σ. In the next section, when we describe the simulation results, we will see how we can manipulate the value of μ and σ as a proportion of π to govern the amount of priming. This amount of priming will serve as a free parameter of the model.

[2] Formally, $\int_{-\sigma}^{+\sigma} \mathcal{N}(\mu,\sigma)\,dx = 0.68 \int_{-\infty}^{+\infty} \mathcal{N}(\mu,\sigma)\,dx$

§ 4.3.3 Simulation & Results

§ 4.3.3.1. **Terminology.—** So far we have been using the term *episode* to indicate the flow of information in a cognitive system and the term *trial* in the same way as it is used in psychological experiments. But in order to describe the simulations, we must refine these definitions and also add a third term, *phase*, to our repertoire. Since we will need to use these terms for specific and disjoint purposes, we provide a brief description of them and their relationship to each other.

- TRIAL. A trial is a single execution of the dynamical system. It commences when the stimuli is input to the system and continues till the system settles down into an equilibrium. The exact time for the system to settle into an equilibrium is determined relative to the time-constant (τ) of the relevant dynamical system. A trial can be either a comprehension trial, or a production trial. The difference is how the input is supplied to the system. A comprehension trial implies that lexical and syntactic choices are not made by the dynamical system (which is the hearer), but by an external speaker. Hence, during a comprehension trial, lexical and syntactic choices are fixed. The mechanism for fixing these choices will be discussed below. In our simulations, a comprehension trial usually serves a *prime* and a production trial could serve either as the *target*, or as a *filler*.

- EPISODE. A comprehension trial followed by a production trial is termed as an episode. The production trial is optional, so that an episode could consist merely of a comprehension trial. The reason for production trial being optional is that we might need to prime the system with a sequence of comprehension trials before simulating a production trial.

- PHASE. When we study the long-term effects of priming, we need to see how a sequence of episodes affect syntactic choice. Such a sequence of episodes consists of a phase. There are two kinds of phases: *training phase*, where the system receives a sequence of primes, and *testing phase* which consists of a sequence of target episodes. Note that a testing phase consists of episodes themselves and not trials directly. This is because we will be using the experimental design of Kaschak and Borreggine (2008) where testing phase consists of a sequence of prime-target pairs. We will discuss their design in greater detail in Section 4.5.3.

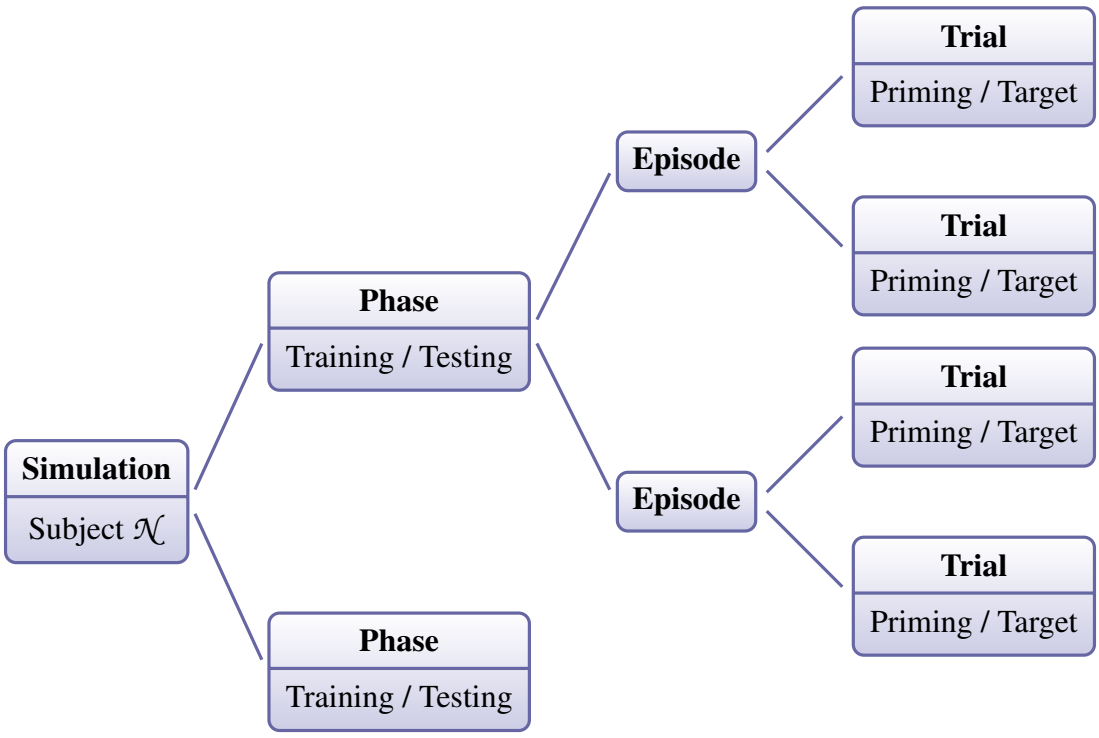


Figure 4.3.5: [Terminological hierarchy] For illustration purposes, each parent node is shown as having only two children nodes. In practice, each parent node will have multiple children. A phase, for example, might consist of ten episodes.

§ 4.3.3.2. **Experiment Design.—** Let us look at the experimental setup for simulating Model ①. This model had two goals: (a) to test whether we can get priming and lexical boost *without* explicit connections between lexical and combinatorial nodes and (b) to test the influence of arousal on priming and lexical boost. Crucially, we do not intend to test the longevity of priming and lexical boost for this model. Therefore, the simulation consists of the prime trial immediately followed by target trial. Figure 4.3.6 shows the experimental setup for model ①. The key thing to note is that the state of the system at the start of the target trial is the same as the state of the system at the end of a prime trial.

Next we need to set the values for the external inputs during priming trials. The priming trial begins with a 'clean slate' system and receives an unbiased external input. This trial simulates comprehension. At the beginning of the priming trial the network was set to the *rest state*, i.e., both the nodes in the WTA layer were turned to the OFF state. In Equation 4.3.1 this corresponds to the initial conditions $E_i = 0, \forall i$. The external inputs during the priming trial are picked randomly from the Gaussian distribution $\mathcal{N}(\mu, \sigma)$. During the priming trials, we would like each of the nodes to be

equally likely to win the competition. As discussed in Section 4.3.2.1, we can express this constraint in terms of the balance between hysteresis and external input being at par – i.e. $\pi = 0.5$. Looking back at Figure 4.3.4, it is clear that we can achieve this if the Gaussian distribution is symmetrical about the boundary between hysteresis and external input. In other words, we need to align the mean with the bifurcation value for $|\Delta K|$: i.e. $\mu = \theta$. Because the Gaussian distribution is symmetric about the mean, the value of the standard deviation, σ, becomes irrelevant in this case. The external inputs were chosen independently for the lexical and combinatorial nodes and the system is simulated till it reaches equilibrium. As the system starts from the rest state during a priming trial, the equilibrium is completely determined by the external inputs.

Now let us look at the target trial. This trial simulates production. We start the target trial by specifying the initial conditions of the system. At the beginning of the target trial, the initial state of each node was set to the final state for the priming trial. Production starts from where comprehension stopped. This means that there is no intervening decay in memory between priming and target and the production dynamics become dependent on comprehension phase, allowing the network to show hysteresis.

Next we specify the external input during the target trial. It is the balance between the hysteresis and external input that decides the extent of automatic priming. When we discussed the architecture for Model ①, we saw that there is a trade-off between automatic and non-automatic (strategic) processes during language production. Syntactic decisions are influenced by the automatic process of priming, but the input from social, pragmatic and inferential premises might constrain this influence of priming. The target trial needs to accommodate these constraints. The architecture of Model ① showed these constraints by making causal link between external input and the *Arousal* module. Now we need to specify how exactly the input from the Arousal module influences the choice of external input during target trial, $|\Delta K_{test}|$.

The sign of the external input is chosen from a uniform distribution such that $P(\Delta K_{test} > 0) = 0.5$ (Equation 4.3.2), so that there is no bias towards any construction. The absolute value of the external input, $|\Delta K_{test}|$, was again chosen randomly from the Gaussian distribution $\mathcal{N}(\mu, \sigma)$. But the mean, μ, and the standard deviation, σ, become a function of π as well as the amount of arousal in the system, $\mathcal{A}$. This means that the difference between the external input for the two nodes is related to the amount of arousal in the system. If the amount of arousal is high, then the difference is larger, making external input dominate over the system's inertia or hysteresis.

Relating external input to arousal makes sense. During production, the external

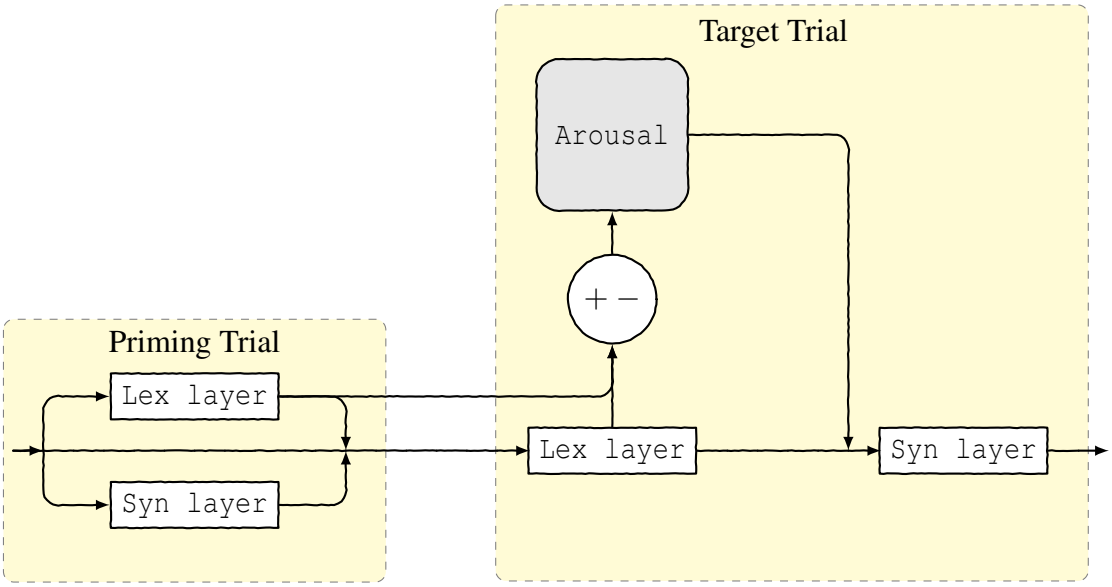


Figure 4.3.6: [Experiment Design] A flow diagram showing the sequence in which different layers are simulated during testing for Model ①. During the priming trial, both layers are simulated in parallel. During the target trial, the lexical layer is simulated before the syntax layer. The arousal is adjusted by comparing the lexemes selected during prime and target. Finally, based on the arousal the inputs to syntax layer are adjusted and the layer is simulated.

input comes from higher level processes. These higher level processes could completely determine syntactic choices (non-automatic processing), or these choices could be left to the syntax module itself (automatic processing). The variable that influences whether the choices will be made within the syntax module, or outside, is arousal. Take, for example, the syntactic choice between a prepositional object and direct object dative. Under some cognitive load, a speaker might choose one over the other based on which was used most recently. But remove this cognitive load and the speaker might favour one of the constructions for contextual or social reasons. This is the trade-off between automatic and strategic processing. The model implements this trade-off through the contrasting forces of hysteresis versus external input. Hysteresis captures the notion of inertia and external input is related to arousal, which, in turn, is related to strategic processing.

Arousal can also be related to lexical repetition. The amount of arousal is a global property of the system. It affects both syntactic and lexical choices. Now, because we are modelling the sentence completion task during the target trial, the lexical choice

has already been made by the experimenter. Subjects can use the verb given in the sentence completion trial to decide the degree of automaticity of their utterance. Or more specifically, they can choose the syntax of their utterance based (either automatically or strategically) on whether the verb given to them during the target trial is the same or different to the prime trial. We can encode these properties of the experimental setup by associating the arousal in the system with the repetition of verb between prime and target.

We assume that the repetition of verb between prime and target trials, leads to a lower level of arousal in the system. We can justify this assumption by considering the definition of arousal in our model. We defined arousal as being inversely related to the degree of automaticity. In terms of the components of automaticity proposed by Bargh (1994), our definition of arousal comes closest to awareness and efficiency. Our assumption is tantamount to assuming that the repetition of the verb leads to low focal attention and therefore triggers efficient processes. This low focal attention also leads to a low level of awareness. In contrast, a novel verb, we assume, contains new information which grabs the focal attention leading to awareness and inefficient processing. While much experimental work needs to be done to confirm (or reject) these assumptions, they allow us to study a possible connection between repetition and the amount of automaticity in the system. We will discuss the consequences of these assumptions in section 4.6.1.

To sum up, the external input for a target trial is picked based on the arousal in the system. Formally, $[\mu, \sigma] = f(\pi, \mathcal{A})$, where $\mathcal{A}$ is the global arousal in the system. The mean and standard deviation of the pdf become a function of π and the amount of arousal, $\mathcal{A}$. This arousal is, in turn, related to lexical repetition. Thus the target trial proceeds in two stages (figure 4.3.6). The first stage simulates the lexical layer by picking external input from $\mathcal{N}(0, f(\pi))$. The second stage simulates the syntax layer by picking external input from $\mathcal{N}(f(\pi, \mathcal{A}))$, where $\mathcal{A} \propto \frac{1}{lexical\ repetition}$. If the lexical layer uses the same verb between first and second stage, the amount of arousal in the system decreases, which, in turn, shifts μ and σ for $p(|\Delta K_{test}|)$. In the current experiment, we assume that μ decreases by a constant k_b in case we have lexical repetition. In other words, the expectation of difference in external inputs for the two nodes decreases by a constant, k_b, whenever the verb is repeated between prime and target. In Figure 4.3.4, this corresponds to moving the Gaussian distribution towards the left by the constant k_b.

§ 4.3.3.3. **Measuring Priming.—** While collecting data from simulating the model, we want to check whether the system shows priming and we want to measure the amount of priming. Therefore, we need a statistic that quantifies the amount of priming shown during a particular simulation. Unlike a psychological experiment, where we specify the outcome of each priming trial in advance, the current simulation determined the outcome of priming trials stochastically (as specified above). In order to measure priming, we needed to compare these stochastic choices during prime and target trials and quantify the results.

Now, we know that priming depends on repetition – an increase in priming would lead to an increase in repetition of a syntactic choice. But this relation is not symmetrical. A large amount of repetition does not necessarily entail large amount of priming. There might be, for example, a bias towards one of the syntactic choices, which would lead to more than a 50% repetition, without priming. We saw in section 4.3.2 that the external stimuli depends on a stochastic variable, ΔK. There, we assumed that the variable is picked from a pdf with mean zero. Moving the mean in any direction can allow us to test the model in cases where there is a syntactic bias towards one combinatorial node. The obvious way to measure priming is to see if a syntactic choice is repeated between a prime trial and a target trial. However, this would mean that the model will show repetition of the preferred structure in absence of priming.

We can circumvent this problem by simulating the system under two different conditions: (i) *Priming condition*: when the initial state of the target trial was influenced by the final state of the prime trial, and (ii) *No-priming condition*: when the initial state of the target trial was reset to the rest state. In other words, we internally switch off priming and measure the amount of repetition. Then we compare this with the amount of repetition in case the priming was switched on. A statistic was derived by performing the simulation for all the subjects[3] and averaging the results:

$$Priming = \frac{N_p - N_{np}}{NSub} * 100 \tag{4.3.6}$$

N_p = Number of subjects that show repetition in priming condition.

N_{np} = Number of subjects that show repetition in no-priming condition.

$NSub$ = Total number of subjects.

In the simulations conducted on this model, each subject undergoes one trial. Therefore this statistic measures priming by trial, rather than by subject.

[3]Each subject is different from the other since the external stimuli, ΔK, is picked independently for each subject, as discussed in section 4.3.2.

To quantify the lexical boost, just like we quantified priming, another statistical measure was developed. This is done by checking if there is greater amount of priming in those cases where subjects receive a lexical boost (i.e. the same lexeme in prime and target)

$$Priming_{rep} = \frac{N_p^{rep} - N_{np}^{rep}}{NBoostSub} * 100 \tag{4.3.7}$$

$Priming_{rep}$ = Amount of priming (in percent) for subjects who received a repeated verb.

N_p^{rep} = Number of subjects (receiving repeated verb) that show repetition in priming condition.

N_{np}^{rep} = Number of subjects (receiving repeated verb) that show repetition in no-priming condition.

$NBoostSub$ = Total number of subjects receiving repeated verb.

Lexical boost can then be calculated as $Priming_{rep} - Priming$. This statistic measures the difference between priming when the verb was repeated and the overall level of priming. If priming is enhanced due to lexical repetition then $Priming_{rep} - Priming$ should give a positive value. Note that this statistic is slightly different from how lexical boost is normally shown in psychological experiments, where the amount of priming is compared between the 'Same-Verb' and 'Different-Verb' conditions. In these results we are not trying to make a quantitative comparison of lexical boost shown by the model and experimental subjects. Rather, we are only interested in accessing whether or not the model shows a lexical boost. The above statistic is sufficient for this purpose.

§ 4.3.3.4. **Results.—** The simulation was run independently for 50 different subjects. Each subject receives a priming trial followed by a target trial according to the experimental design shown in Figure 4.3.6. The results of the simulation are shown in figure 4.3.7. Along the y axis, the figure shows amount of activation of a node at the end of a trial, i.e. when the system has reached equilibrium. Circles represent the activation of a PO node, while diamonds represent the activation of a DO node. Along the x axis, the figure shows the subjects.

We can observe from the figure that the activations of the two nodes are polar opposites. For a particular subject, if the PO node achieves maximum activation, the DO node is completely switched OFF and shows zero activation. We can find the amount of repetition by comparing the results from the priming and the target trials. If a node that wins the competition during the priming trial also wins the competition during the target phase, then we have repetition. We plug in these results into Equation

4.3.6 to obtain $Priming = 20\%$. These results were caluculated for $\mu = \theta$, i.e. the mean of the Gaussian distribution was situated at the point of bifurcation. Since hysteresis dominates half of the times, it is not surprising that the model shows priming.[4]

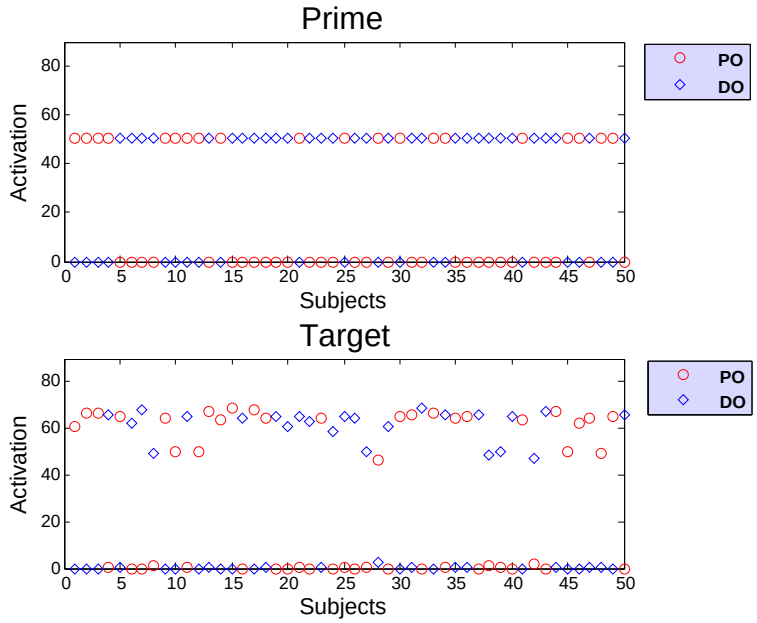


Figure 4.3.7: [Results] The results for simulating Model ① for fifty subjects. The x axis shows the subjects and y axis shows the activation of the PO and DO node at the end of each trial.

We also checked whether the model showed an increase or decrease in priming with change in external stimulus. To do this, we varied μ and σ. This is equivalent to changing the shape of the Gaussian in Figure 4.3.4 and moving it along the axis. When we increased μ to $(\theta + 5)$, and kept σ to a very low value of $(\theta/50)$, *Priming* became 0%. External input becomes the governing force, removing the effect of hysteresis completely. On the other hand, when we moved μ in the other direction to $(\theta - 5)$, *Priming* became 44%. Thus the model behaved as we had expected – showing priming based on the balance between external input and hysteresis.

Next, we tested the network for lexical boost by setting k_b to a positive value. Until now, the global arousal in the system did not affect the value of external stimuli. But when we set k_b to a positive value, the difference in external input decreases when the

[4]If anything, it is surprising that the model shows a priming of 20% and not 50%. This is due the fact that our statistic compares the repetition under the priming and no-priming conditions. Since there is repetition during the no-priming condition as well, the overall amount of priming shown by this statistic is diminished.

system shows low arousal (Section 4.3.3.1). In other words, when the verb is repeated between prime and target trials, the new expected value of difference between external inputs is reduced by k_b: $|\Delta K_{test}^{new}| = |\Delta K_{test}^{old}| - k_b$.

We reset the value of $\mu = (\theta + 5)$ and $\sigma = \theta/50$, a setting of parameters that previously gave us no priming. We also set $k_b = 6$, so that it can just overcome the external input. When the model was tested with these parameters, *Priming* indeed increased to 12%. Whats more, the value of the statistic $Priming_{rep}$ – i.e. the priming for those subjects which received repeated verbs between prime and target trials – became 23%. Thus subjects who received the repeated verb show a large amount of priming taking the overall level of priming up from 0% to 12%.

This pattern of results remained true for various values of the parameters μ, σ and k_b. The model showed both priming and lexical boost, contingent on values of these parameters. By varying the parameters, we can confirm that the reason that the model shows priming is indeed the the balance between hysteresis and external input and it shows lexical boost due to the input from the arousal module. Thus these simulations serve as a proof of concept for our predictions of the reasons behind structural priming and lexical boost. Of course, the value of these results rests on a set of assumptions about the role of hysteresis and arousal in linguistic processing. We discuss these assumptions and other the consequences of our results in section 4.6.1.

4.4 Model Two

Model ① challenges the idea of overt associative links between lexical and combinatorial nodes. It replaces these overt associative links with a module governing the degree of automaticity in the system. We saw in the simulations that such a model can lead to lexical boost. Therefore model ① provides a proof of concept for a complex relationship between lexical and syntactic representations. In this section, we revert to the simpler possibility that information flows between lexical and syntactic representations through a set of associative links. These links are simple to implement and help us explore the longevity of the association between lexical and syntactic information. A complete model would consider the interplay between flow of information due to the degree of automaticity in the system and due to associative connections between lexical and syntactic representations. However, evaluating such a model would require experimental data to be collected on the effect of automaticity on syntactic decisions, which is lacking at the moment. Therefore, we will not endeavour to construct such

a model. Instead, we pursue the idea of direct associative links between lexical and syntactic representations and explore whether such links can lead to structural priming and lexical boost.

Specifically, we consider the possibility that the links between lexical and combinatorial nodes are present, but do not undergo long-term learning. That is, these links simply function as a flow of activation from one representation to the other. This flow of activation might be gated (i.e. the links become active under certain circumstances and broken under others), but they does not learn the statistical association between lexical and combinatorial information.[5] Our next model explores whether we can get priming and lexical boost when the links between lexical and syntactic information do not show this long-term statistical learning.

This model challenges the notion that the causal link between lexical and syntactic information is necessarily formed through *long-term* learning, or that these links record statistical information over a series of episodes. Chang et al. (2006) predict that the same learning mechanisms that are involved in long-term language acquisition could also lead to priming effects. Their model learns the statistics of stimuli over a series of episodes and incrementally accommodates each stimulus into the model. Model ② tries to put forward a contrasting hypothesis: Short-lived memory traces can also reproduce experimental findings related to short-term priming and lexical boost. We discuss the consequences more fully after presenting the results.

§ 4.4.1 Architecture

Figure 4.4.1 shows the network architecture of our next model. The nature of connections between nodes in the same layer is the same as in model ① – inhibitory connections allowing for competition. However, instead of having no links between nodes in the two layers, this model assumes that any pair of nodes in the two layers are connected through a module of short-term memory (STM). This module remembers the association between a lexical and syntactic information for a short period of time. It contains nodes that can remember the conjunction of lexical and syntactic properties of a sentence. The reader would recall (from Section 4.2.3) that remembering such conjunctions involves forming the binding between the two properties. Since we pre-allocate the nodes that are used for storing each conjunction, our model implements

[5] In the strictest sense, this opening and closing of gate might be (and is) governed by the input and therefore counts as a kind of learning too. But here we are reserving the term *learning* for learning the statistical patterns (e.g. correlation) between lexical and combinatorial information.

static binding. The short-term memory layer is nothing but a collection of binding nodes. The number of binding nodes is equal the total number of combinations of lexical and syntactic nodes.

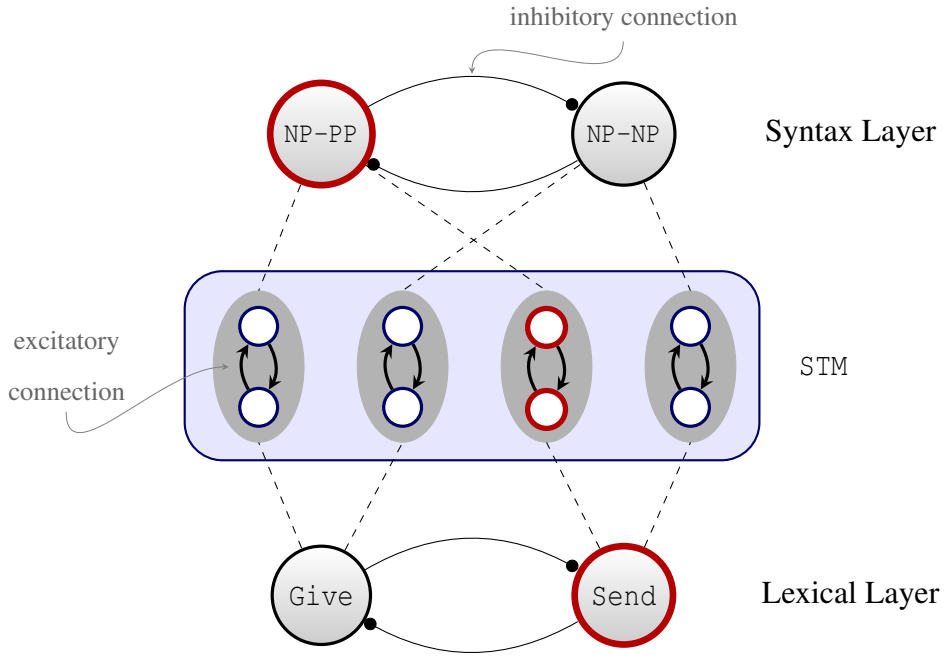


Figure 4.4.1: [Model ②] Syntax and lexical layers consist of mutually inhibitory nodes. STM consists of a set of binding nodes (shown as gray ellipses) with each binding node containing two mutually excitatory nodes. A binding node is activated when nodes connected to it in both the lexical and syntax layer are activated.

As shown in the figure, each binding node connects to two nodes – one in each layer. Node 1 for example is responsible for retaining the binding between a PO and Give. If both of these nodes are activated during a priming trial, then binding node 1 gets activated, thereby memorising the association. The memory is short-term and is wiped out completely at the end of each trial.

Our goal in this study is to investigate the lexical influence on syntactic choice. We are not interested, for our present purposes, in the role of syntactic information on lexical choice. For this reason, the connections between the STM and the two layers are asymmetrical. Activation flows only one way – from the lexical layer to the syntactic layer. We call these connections *control connections* and the exact detail of the flow of information on these connections is discussed when we describe the simulation (see Section 4.4.3).

§ 4.4.2 Formal Description and Dynamics

Model ② is an extension of Model ① and its formal description can also be seen as an extension of the formal description of the previous model. It still contains the competitive networks of mutually inhibitory nodes. However, in addition, it also contains some additional dynamical systems. In this section, first we describe the dynamical subsystems and then we outline how these subsystems can implement a key goal of the second model – decay in memory. Once we have described these systems formally, we will be ready to test them. The method of determining the external stimuli remains the same as the previous model and will not be described again.

§ 4.4.2.1. **Dynamics.—** Model ② extends Model ① in two ways: it introduces a short-term memory module between *Layer*1 and *Layer*2 and it introduces decay in memory through adaptation. In this section we formalise the notion of short-term memory and in the next we formalise decay in this memory. The dynamics for winner-take-all competition within a layer is inherited in this model from Model ①.

As we saw in Section 4.2.3, the short-term memory module consists of a number of binding nodes that retain the association between nodes in *Layer*1 and *Layer*2. One way to encode the association of two nodes is by simply setting an attached flag whenever two nodes are activated together. However, such a mechanism will not be consistent with our overall paradigm of using dynamical systems. Additionally, there is no straightforward way to implement forgetting using such a mechanism. In contrast, the additive model that we discussed above can show both short term memory and decay. Therefore, we implement the binding nodes using such an additive model.

Specifically, each binding node is implemented as a network of two mutually excitatory nodes (figure 4.4.2). It has previously been shown (e.g. H. R. Wilson (1999)) that such a pair of mutually excitatory binding nodes can, just like the WTA pair, show hysteresis. Once activated beyond a particular point (i.e., driven to the ON state), both nodes will need to be sufficiently inhibited by external input before they are turned OFF. The dy-

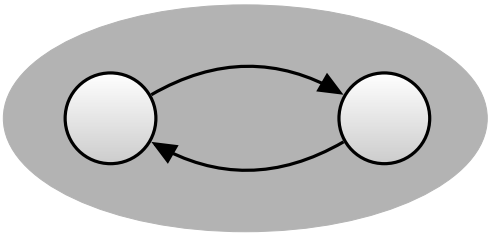


Figure 4.4.2: [Binding node] A binding node is a dynamical system consisting of two mutually excitatory nodes.

namics of each node in this pair is given by:

$$\frac{dE_i}{dt} = \frac{1}{\tau_{stm}}(-E_i + S(K_{stm} + cE_j)) \tag{4.4.1}$$

where E_i and E_j are activations of the two mutually excitatory nodes comprising a binding node, τ_{stm} is the time constant for the rate of change of this activation, K_{stm} is the external input, and $S(x)$ is again the Naka-Rushton function. In contrast to the winner-take all dynamics of Equation 4.3.1 (page 112), each node in this system gets positive feedback from the other node. Also note the missing summation function since we assume that this network consists of exactly two nodes, while the WTA network consisted of two or more nodes.

Comparing equations 4.4.1 and 4.2.3 (page 95), we can see that each excitatory node is an additive model, with each node getting positive feedback from the other. So it would come as no surprise that this network has three states of equilibrium, two nodes and one saddle point – just like the WTA network. However, unlike the WTA network, the stable equilibriums occur at $(0,0)$ and (ν,ν) – where ν is a positive constant (see Figure 4.4.3). The exact value of ν is currently not of interest to us, but the interesting observation is that both nodes have the same amount of activation in either of the stable states. Both nodes either switch ON (ν,ν), or OFF $(0,0)$. Therefore, the state of the network (i.e. the binding node) is stored mutually in the two nodes. If both the nodes are in the ON state, then a conjunction is present, otherwise it is absent.

§ 4.4.2.2. **Adaptation.—** We picked the above dynamical system to represent binding nodes not just because it could encode conjunctions, but also because this system can show forgetting. The conjunctions are implemented using the stable nodes and forgetting is implemented through loss of this stability. In between the two stable nodes, $(0,0)$ and (ν,ν), lies an unstable saddle (κ,κ). We have seen this pattern before. It occurred when we were looking at mutually inhibitory nodes (figure 4.2.7) and saw that the isoclines for such nodes intersected in three places. There as well, an unstable saddle was sandwiched between two stable nodes. Before that, we saw the same pattern in Figure 4.2.4 (page 99), when we had a first look at adaptation. There we looked at the isoclines for an additive model. They intersected in three places, provided an adaptation parameter was below a certain value. As the adaptation parameter increased, the unstable saddle coalesced with the stable node leading to a loss in memory. In this section, we are interested in (a) specifying how the adaptation parameter might change with time, (b) identifying a system variable that can act as the adaptation parameter and

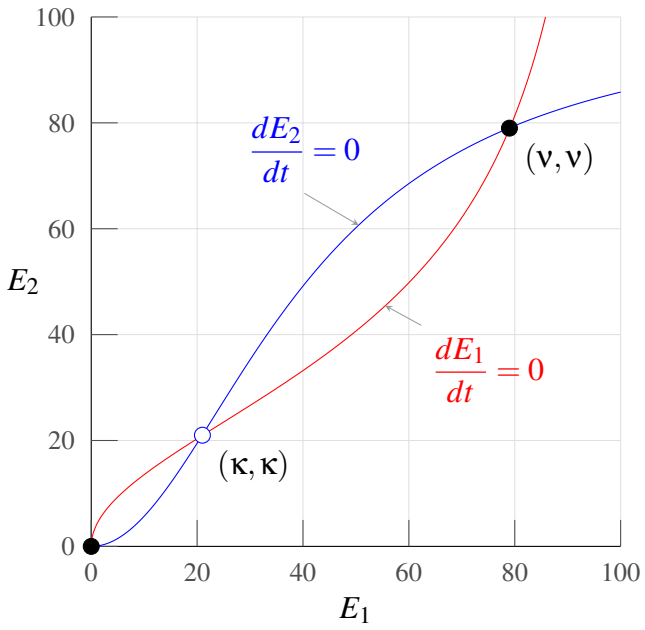


Figure 4.4.3: [STM Isoclines] Isoclines for mutually excitatory nodes show that, at equilibrium, either both nodes will turn ON (ν, ν), or both nodes will turn OFF $(0, 0)$. The intermediate equilibrium (κ, κ) is an unstable saddle point (H. R. Wilson, 1999).

(c) understanding how the change in this parameter could lead to forgetting in STM and WTA nodes.

Let us first consider a winner-take-all network. The memory of such a network lies in remembering which of the two nodes received greater external input during the priming trial. The system is capable of remembering such an event because each event corresponds to a stable equilibrium and each stable equilibrium is the polar opposite of the other stable equilibrium. One can, therefore, think of decay as a mechanism that makes the system move away from this polar nature, either by causing the stable equilibriums to disappear, or by bringing them close to each other (thereby diffusing the polarity). Therefore, the adaptation parameter will need to be changed incrementally and with each incremental adjustment, the system should move away from these stable states. Each node in the WTA (or STM) network is an additive model:

$$\frac{dx}{dt} = -x + S(\rho + cx)$$

As we discussed in Section 4.2.1.2, this additive model has two bifurcation parameters: the external input, ρ and the feedback, c. We have already seen how the external input might push such a model over a bifurcation point leading to a change in its qual-

itative behaviour. This same qualitative change can be achieved using the feedback parameter, c. We cannot use the external input, ρ, to account for adaptation, since the value of this variable is governed by the input stimuli. However, there is a possibility that we can use the feedback parameter c to perform the role of adaptation.

Here we run into a problem. If c is to serve as an adaptation parameter, then we will need to make incremental changes to it. Feedback to each node comes from the input connection from the other node. In the WTA network these connections are inhibitory and in the STM network they are excitatory. However, the strength of connections between nodes in a layer needs to be symmetrical and fixed because we do not want to build any Hebbian-like long-term learning into Model ②. But this means we cannot adjust the strength of these connections to take care of adaptation.

The solution to this problem comes from the study of cortical gain control circuits. H. R. Wilson and Humanski (1992) show that if a cortical circuit has divisive feedback[6], then changing the feedback is equivalent to changing the semi-saturation constant, $\hbar$, of the neuron's response. The reader would recall that the semi-saturation constant is the value of input stimuli for which the Naka-Rushton response function reaches half its peak-value (equation 4.2.5). For our purpose, $\hbar$ gives us a parameter that we can relate to a neuron's feedback. Crucially, H. R. Wilson (1999) show that this variable can act as the adaptation parameter in Equation 4.2.4. As the parameter changes with time, the stable nodes in figures 4.4.3 and 4.2.7 will approach the unstable saddle, eventually leading to a bifurcation and a loss in the system's memory.

Model ② consists of two kinds of connections – mutually inhibitory (WTA), and mutually excitatory (STM). Each of these dynamical systems (given by 4.3.1 and 4.4.1) is extended to adopt the changing parameter $\hbar$. For the STM network, this is done by replacing $\hbar_1$ with $(\hbar_1 + A_1)$ and $\hbar_2$ with $(\hbar_2 + A_2)$, and adding the equations for change of A_1 and A_2. Here A_1 and A_2 are the amount of adaptations to $\hbar_1$ and $\hbar_2$ respectively. The complete system becomes:

$$\begin{aligned} \frac{dE_i}{dt} &= \frac{1}{\tau_{stm}}(-E_i + S(K_{stm} + cE_j)) \\ \frac{dA_i}{dt} &= \frac{1}{\tau_a}(-A_i + \alpha E_i) \end{aligned} \tag{4.4.2}$$

where τ_a is the time constant for adaptation and α is the saturation constant for A_i – i.e., it governs the maximum values of A_i, as a fraction of E_i.

[6] In a cortical circuit, divisive feedback means that the input signal is divided by a portion of the neuron's response. Unlike subtractive feedback, divisive feedback depends nonlinearly on a neuron's response.

For the WTA network, we make a slight modification. Instead of having different rates of adaptation of E_1 and E_2, we have the same rate of adaptation, based on the sum of the two activations, so that the complete system becomes:

$$\begin{aligned} \frac{dE_i}{dt} &= \frac{1}{\tau}(-E_i + S(K_i - c\sum_{j\neq i} E_j)) \\ \frac{dA}{dt} &= \frac{1}{\tau_a}(-A + \alpha\sum_j E_j) \end{aligned} \tag{4.4.3}$$

We can justify using the same rate of adaptation for both nodes in a WTA network from both a computational perspective and a behavioural perspective. From the computational perspective, having separate adaptation rates for E_1 and E_2 leads the network to an asymptotically stable limit cycle. This means that E_1 and E_2 start showing oscillations. As adaptation proceeds, the activation of one of the ON node starts to decrease and that of the OFF node starts to increase. This continues till the previously OFF node has maximum activation (becomes ON) and the ON node has zero activation (turns OFF). This is not phenomenologically desirable as it would mean that we get "reverse-priming" in case the testing phase begins at a time when E_1 and E_2 have reversed their activation during an oscillation. There is no such evidence of an oscillating amount of priming (see, for example, Pickering and Ferreira (2008)). Also, from a behavioural perspective, it makes sense to have the same adaptation rate in both the nodes as the nodes are completely symmetrical. This adaptation rate is a function of the total activation in the network $(E_1 + E_2)$, rather than activation of a particular node. However, it must be said that the adaptation rate in this case becomes a non-local behaviour – i.e. its value does not depend simply on the local variables of the node, but on the global state of the system.

§ 4.4.3 Simulation & Results

§ 4.4.3.1. **Experiment Design.—** Just like Model ①, the simulation consists of a sequence of two trials: the *priming trial* followed by the *target trial*. And just like Model ①, the external inputs K_1 and K_2 are selected stochastically. During the priming trial, the initial state of all nodes is set to the rest state ($E_i = 0, \forall i$). ΔK_{prime} is chosen in an identical manner to Model ① from the pdf $\mathcal{N}(\mu = \theta, \sigma)$ and equilibrium state is calculated by simulating the dynamical equations of both layers. Once the network has settled, the bindings are stored in the STM (see figure 4.4.4). This is done by simulating the dynamical equations of the mutually excitatory networks and a constant external

input. This external input is positive (and above threshold) if both nodes connected to the binding node are active (ON), and zero otherwise.

The target trial simulates production. We are interested in syntactic choice and therefore, this target trial can be seen as simulating the sentence completion task in which the subject is already given the verb, but can choose one of two possible grammatical structures to complete the sentence. The lexical layer is simulated first and receives a large external input ($\mu \gg \theta$), ensuring that the lexical decision is made solely using external stimulus. After the nodes in the lexical layer have achieved equilibrium, the syntax layer is simulated. The layer receives a random external input (ΔK_{test}) that is not biased towards any node. The amount of priming is varied by varying $p(|\Delta K_{test}|)$, as described in Section 4.3.2.

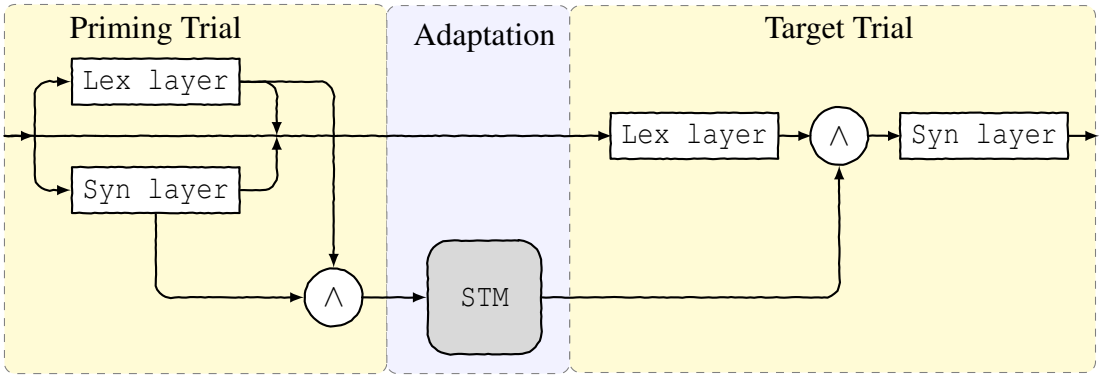


Figure 4.4.4: [Experiment Design] Flow diagram for simulating Model ②. Unlike the simulation for Model ①, the priming and target trials are separated by an adaptation period. The short-term memory is updated after the priming trial and forms an input to the syntax node during target trial. $\wedge$ represents the process of finding conjunction. It is used to both bind the representations during a priming trial and unbind them during a target trial.

Unlike Model ①, this model has explicit connections between the lexical layer and the syntactic layer, via the short-term memory. Each node in the syntactic layer receives input from STM based on the following expression:

$$\begin{cases} K_{stm} & if \quad E_{stm} \times E_{lex} > 0 \\ 0 & otherwise \end{cases} \tag{4.4.4}$$

where K_{stm} is a constant amount of input that a node in syntactic layer gets from the STM, provided the condition on the right-hand-side is met. E_{stm} and E_{lex} are the activations of the binding node and the lexical node connected to it, respectively. The

condition $E_{stm} \times E_{lex} > 0$ ensures that a syntactic node receives input from only those binding nodes that are themselves active *and* are connected to active lexical nodes. For example, to calculate the input from STM to the PO node, the model tested whether $E_{Give} \times E_{stm1} > 0$, or, $E_{Show} \times E_{stm2} > 0$. Thus this condition ensures that a PO node gets input from the STM layer only if a connected binding node was activated during the priming trial (hence, $E_{stm1} > 0$ or $E_{stm2} > 0$), *and*, the lexical node linked to this binding node is active during the target trial. We see that during the target trial, there is a causal flow of activation from lexical to syntax layer via the binding nodes.

It should be noted that the dotted connections between the nodes in Figure 4.4.1 (page 126) – i.e., the connections between the STM and the two layers – are not excitatory or inhibitory connections like the ones within each layer. Instead, the flow of activation along these connections is dependent on a logical condition (expression 4.4.4). It is for this reason that we call these connections *control connections*.

There is another way in which simulating this model is different from the previous one. The target trial immediately follows the priming trial in Model ①. Here, the two are separated by a period of adaptation. While explaining the simulation of the target trial above, we did not state the initial conditions for the nodes in the WTA layers. These initial conditions are a key difference in the previous simulation and this one. In Model ① the initial state of the dynamical system during the target trial was just the final state of the priming trial. In the current model, once the priming trial is over, the system keeps running under adaptation conditions (equations 4.4.2 and 4.4.3) before the target trial.

During the adaptation period, the nodes in STM network received no external input while the nodes in WTA layers received a constant external input. This difference in external input for the two types of nodes is required by the nature of their connectivity. Since the connections within a binding node in STM are mutually excitatory, the network is able to retain its activation in the absence of any external input. However, the connections in the WTA network are mutually inhibitory. In the absence of an external input, these nodes rapidly pull down each other's activity. The nodes soon approach the rest state, losing their memory. Looking back at equation 4.4.3, this behaviour would mean that the activations start decaying according to the time constant τ, rather than τ_a, which was meant to be the adaptation time-constant.

We can prevent this rapid decay in WTA memory by maintaining an external input during the adaptation period. However, maintaining the same external input as the stimuli is not behaviourally desirable as (a) it is biologically implausible, and (b) it

would not let the system adapt, defeating the purpose of adaptation period. Instead, we can maintain a neutral external input, which does not favour any particular node (i.e. $\Delta K = 0$). Such an input allows the system to adapt according to the adaptation time-constant, τ_a, and does not interfere with the network's memory. The absolute value of each external input during adaptation, K_{adapt}, is set to the mean of the two inputs K_1 and K_2 received during the priming trial.

Adaptation carries on for a pre-determined amount of time (and not till the system has reached equilibrium). At the end of this period, the initial state for the target trial is chosen as the final state of the system during adaptation. Activation of both WTA nodes and STM nodes changes during adaptation period. This means the model will show variable behaviour depending on the duration of adaptation.

§ 4.4.3.2. **Results.—** Before we look at the results of the simulation, let us review the hypotheses of this model. The simulation tested the model for three kinds of behaviour:

(i) Whether or not the model shows priming.

(ii) Whether or not the model shows lexical enhancement of this priming.

(iii) How do priming and lexical boost vary with change in duration of adaptation?

Our prediction is that the model should show both priming and lexical boost. This model, just like the last one, shows WTA dynamics. These dynamics exhibit hysteresis which lead to priming in Model ①. Therefore, we predict that the model should show priming. Furthermore, the combinatorial nodes in the current model receive an input from the lexical layer (via the STM module). Repeated lexemes between prime and target trials should lead to extra input for the combinatorial node that appeared during the priming trial. Thus, we expect to see a lexical enhancement of structural priming.

We also expect that the amount of priming and lexical boost will decay with increase in adaptation time. This can be seen by looking back at equations 4.4.2 and 4.4.3. As the adaptation period progresses, the adaptation variable A_i increases, leading to a decrease in the activation for the nodes. This, in turn, drags the stable system towards the saddle point. The longer the adaptation period, the closer the system will get to the unstable saddle and the more likely it is to lose its memory.

Figure 4.4.5, shows results from a typical simulation run on 50 independent subjects. Like the results of model ①, this figure plots the output of both the priming and target trials for the syntax layer. Model ① contained only the lexical and syntax layers,

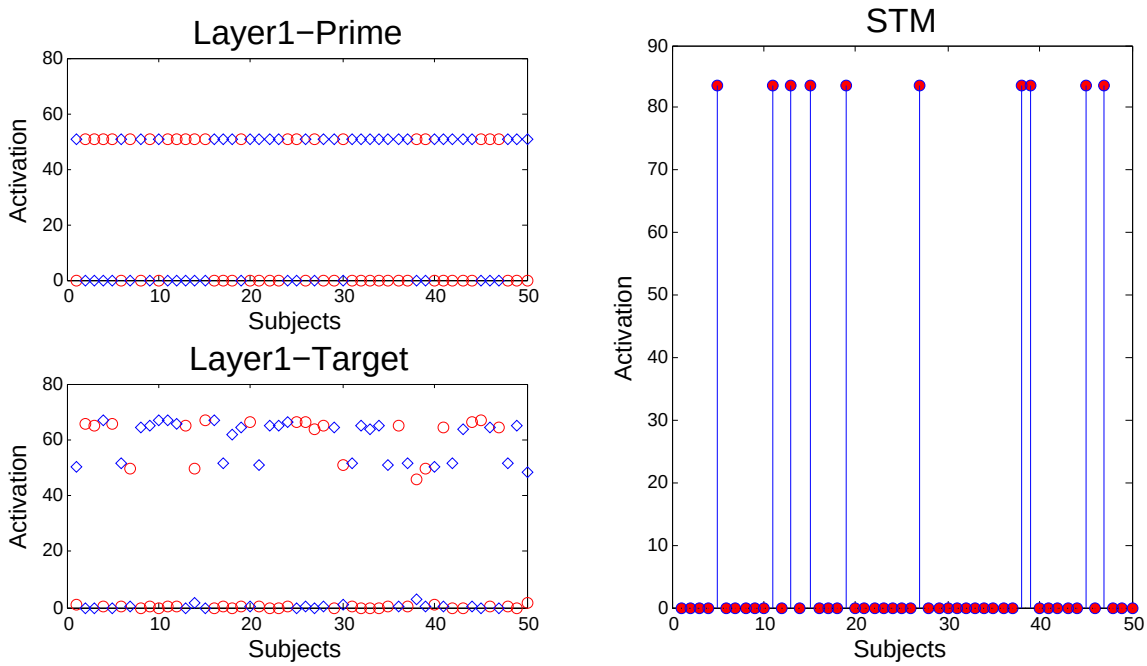


Figure 4.4.5: Results for simulating Model ② on 50 subjects. The x axis shows subjects and y axis shows activation. The two panels on the left show activation for PO (circle) and DO (diamond) nodes. The panel on the right shows the state of one (out of the four) binding nodes, for all 50 subjects. If a binding node is active (i.e. activation is more than 50), then it passes the activation from the associated lexical node to the syntax node.

but this model also contains a connecting layer of binding nodes. Therefore Figure 4.4.5 also shows the activity in the binding nodes alongside. Also like the previous model, these results are contingent on the values of the parameters μ and σ which determine the external input, drawn from $p(|\Delta K_{test}|) = \mathcal{N}(\mu, \sigma)$. For the results shown in figure 4.4.5, we had set $\mu = (\theta + 0.5)$ and $\sigma = (\theta/20)$. This means that the system is marginally in favour of the external input as the mean of the Gaussian distribution in figure 4.3.4 (page 115) is marginally to the right of the bifurcation point θ. In addition, we set the adaptation time between prime and target trials to be 0ms – i.e. there was no adaptation. For these values of the parameters, we obtained $Priming = 20\%$ and $Priming_{rep} = 18.5\%$. Thus, the system seems to show priming but since $Priming$ and $Priming_{rep}$ are almost the same, it shows no lexical boost.

The reason for the lack of lexical boost is that the syntactic layer in the system received no input from the binding nodes – i.e. we had set $K_{stm} = 0$. Next, we wanted to see how the system behaves when we changed the value of this constant – i.e. activated the connection between binding nodes and syntactic layer. We also wanted to see the

Table 4.1: Results for Model ②. The first two columns show values for different parameters. We keep μ and σ fixed and vary K_{stm} and the adaptation time.

K_{stm}	Adapt time	*Priming*	$Priming_{rep}$	Lex boost
$K_{stm} = 0$	0ms	20%	18.5%	✘
$K_{stm} = 2$	0ms	39%	51.2%	✔
$K_{stm} = 2$	1000ms	32%	38.2%	✔
$K_{stm} = 2$	2000ms	34%	43.5%	✔
$K_{stm} = 2$	3000ms	18%	20%	✘
$K_{stm} = 2$	4000ms	13%	12.5%	✘
$K_{stm} = 2$	8000ms	11%	12%	✘

behaviour of the system when the adaptation time was steadily increased. In particular, we wanted to observe how priming and lexical boost change as the lag between priming and target trials is increased. Testing the adaptation time requires setting another parameter of the model – the adaptation time constant τ_a (equations 4.4.3 and 4.4.2 on page 130). We assumed the value of 4000ms for this adaptation time constant. This value is chosen so that it is much larger than τ and τ_{stm}, the time constant for change in activation in syntax and STM layers. Crucially, this adaptation time constant is the same for both the syntax layer and the STM layer – i.e. we assumed that similar kinds of decay mechanisms operate in the representational layers and the binding nodes. The reason for making this assumption is that we wanted to explore whether the same decay mechanisms in the two memory systems can lead to different properties of decay in structural priming and lexical boost. The results from simulating Model ② with a range of values for adaptation time are shown in Table 4.1.

From Table 4.1, we can observe that the model is capable of showing both priming and lexical boost. This lexical enhancement of priming is contingent on input from the STM module. In the absence of this input – $K_{stm} = 0$ – the model shows no lexical boost. We also observe that the amount of priming reduces steadily as the adaptation period increases. Over a long period of time (Adapt time = 8000ms) the priming shows saturation, decreasing to around 11%. The lexical enhancement of this priming also decreases steadily with increase in adaptation time. But in contrast to priming this lexical boost decreases suddenly to zero between 3000ms and 4000ms. Thus, these results not only confirm that Model ② is capable of showing structural priming and

lexical boost, they also show that the same internal decay rate in memory (τ_a), can lead to different behavioural decay properties. We discuss the reasons and significance of these results in Section 4.6.2.

4.5 Model Three

In the first two models, we experimented with the nature of connections between lexical and combinatorial nodes. The first model had no explicit links between the two representations and the second model had links that stay active over only a short period of time. In Model ③, we make it possible for the links to remember associations between lexical and combinatorial nodes over a sequence of episodes. That is, we provide the network with long-term memory. Specifically, we build in incremental Hebbian learning. Unlike the network in Model ②, this network is able to extract statistical patterns of association between the two representations. However, unlike the network presented in Chang et al. (2006), Model ③ uses *unsupervised* learning. That is, there is no model for the correct response that the simulation tries to internalise. This, in turn, means that learning is not error-based. The consequences of this scheme are discussed in Section 4.6.

§ 4.5.1 Architecture

Figure 4.5.1 shows a schematic diagram of model ③. This schematic representation is the same as that for model ②, except for the arrows attached to the input of the nodes.[7] These arrows depict the external inputs that the nodes receive. During comprehension, this input is assumed to be from feed-forward connections coming from either lower levels of processing or afferent connections from the sensory system. During production, this input could come from other cognitive systems, such as meaning and attention. In addition to this, nodes in layer 1 and 2 also receive a fixed input from binding nodes in the short term memory.

The reason why external inputs are explicitly represented in figure 4.5.1 and not in Figure 4.4.1 is that these external inputs are used to store long-term memory in model ③. Every time a node gets activated, its external input is incremented. This is

[7]The number of words in the lexical layer has been increased to four in place of two in the previous model. We made this change because this model will be used to test the stimuli from Kaschak and Borreggine (2008) which requires at least four verbs. However, this is not a major change and Model ③ can easily be changed to use two verbs like the previous models without any loss of generality.

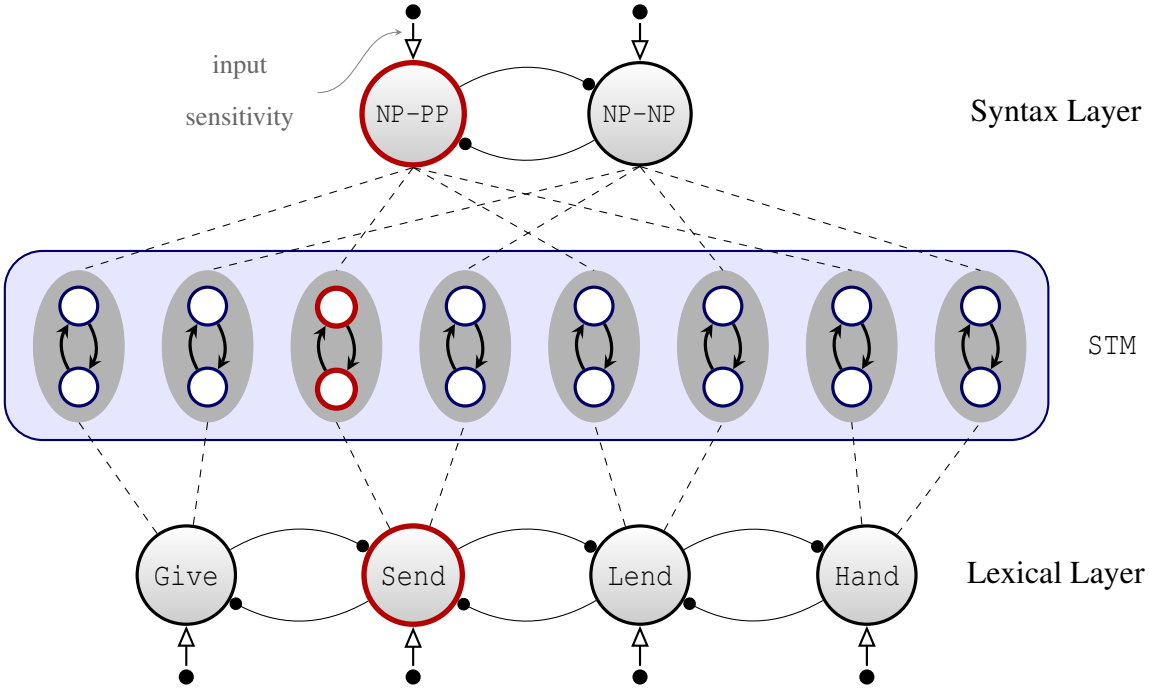


Figure 4.5.1: [Model ③] All nodes in a layer are connected with inhibitory links and each node also has control connections to binding nodes in the STM layer. (For representational convenience we show inhibitory links only between neighbouring nodes, but each layer is, in fact, fully connected. The formal description below will give a more accurate view of network connectivity.) The input sensitivity of each node (shown as incoming arrow with weight) varies with each episode.

true for both nodes in the representational layers and the binding nodes in the short-term memory module. There are various other ways in which long-term memory can be implemented in the model. We will discuss in the next section why we choose to implement long-term learning through incrementally adjusting the input to nodes.

§ 4.5.2 Formal Description and Dynamics

Model ③ inherits its dynamics from Model ② – two WTA layers connected with an STM module consisting of binding nodes. Both WTA and STM show adaptation with time, as discussed for model ②. The key addition to this model is that of long-term cumulative learning. In this section we present a formal description of how such a cumulative learning can be encoded and how to select the parameters associated with this learning.

§ 4.5.2.1. **Incremental learning.—** Model ③ extends Model ② by performing long-term learning. While in Model ② we were content to use the current state of the system as the memory for its recent past, in this model we explicitly record information of each passing episode. Two kinds of information are recorded:

(i) The association between lexical and combinatorial information. This information is encoded in the binding nodes.

(ii) The combinatorial and lexical choice made during the target trial. This information is encoded in the winner of the comptetion in each WTA layer.

Incremental learning can be seen as an extension of the trailing activation account (Pickering & Branigan, 1998), which claims that priming is due to traces of residual activation in the language processing system. Incremental learning proposes that these traces can accumulate over time, so that more frequent constructs are more likely to be chosen over low frequency constructs, given the same amount of attention. Thus, a system that shows incremental learning is sensitive not just to the most recent episode, but also to the long-term history of the inputs presented to the system.

Since this incremental learning is simply an accumulation of trailing activation, it is in contrast with goal-directed learning mechanisms. The learning algorithm for our system does not try to model the environment – it simply records traces of activation.[8] This is an important distinction – while a learning algorithm that tries to model the environment is concerned with the goal of the learning first and cognitive implementation second, a learning algorithm that records traces of activation places the emphasis on cognitive implementation. The difference is a matter of perspective. A goal-directed learning algorithm looks at the cognitive system from the outside, inwards, and tries to find an implementation that models the environment. A learning mechanism that records the activation looks from the inside, outwards, and finds a property of the environment that is encoded through forming such a record.

A cognitive system could record a series of episodes in two possible ways: it could record each of these episodes separately as an *exemplar* (e.g., Goldinger, 1998), or it could collapse information spread across multiple episodes into a single variable and use each episode to update the variable. The first algorithm uses an episodic memory to retain episodes, while the second one uses a prototype-based semantic memory. In this chapter we implement long-term learning as a prototype-based memory. Each

[8] Inevitably, this accumulation of traces of activation models the frequency of stimuli in the environment – however, this can be seen as a by-product of the learning algorithm, rather than its goal.

episode is used to incrementally update a variable that records the system's response to the input. Specifically, we use a Hebbian-like *unsupervised* learning algorithm. The system has no explicit teacher and the training data consists of a set of inputs without any corresponding target values. The learning algorithm makes fixed increments to its semantic memory based on each episode.

There are a number of variables in the model that can record the effect of each episode:

- PEAK ACTIVATION OF EACH NODE. The peak activation of each node is governed by the variable M in the Naka-Rushton function

$$S(x) = \begin{cases} \dfrac{Mx^N}{\hbar^N + x^N} & for \quad x \geq 0 \\ 0 & for \quad x < 0 \end{cases}$$

  One way to record the effect of each episode is to increase the value of this peak activation incrementally when the node gets activated. This way a node that has been activated more frequently will have higher activation as compared to a node that gets activated less frequently. This would also mean that for the same amount of external input, the node that gets activated more frequently will show a larger response to the input and as a result dominate the other node.

  Although this mechanism allows us to model the effects of repeated activation of a node, it raises questions about the neural plausibility of the model. Neurons show a lot of random activity that (until now) cannot be associated with any behaviour. This random neural activity, or *neuronal noise*, interferes with the value of the peak activation of a node. If, for example, a node in our system is implemented through a population of neurons, then the peak activation of this population will not be a constant. Rather, it will show a large variation because of neuronal noise. This variation in peak activation will mean that small increments to this activation after each episode cannot be seen as an evidence for learning. Therefore, using peak activation is not desirable for encoding learning in our system.

- STRENGTH OF EXCITATORY OR INHIBITORY CONNECTIONS. So far we have assumed for both STM and WTA networks that the connections between any two nodes are symmetric. But this need not be the case. If we make one of the connections stronger than the other, it alters the equilibrium states of the network. This can be seen in figure 4.5.2, where the inhibitory connection from

the first node to the second is stronger than the inhibitory connection from the second node to the first.

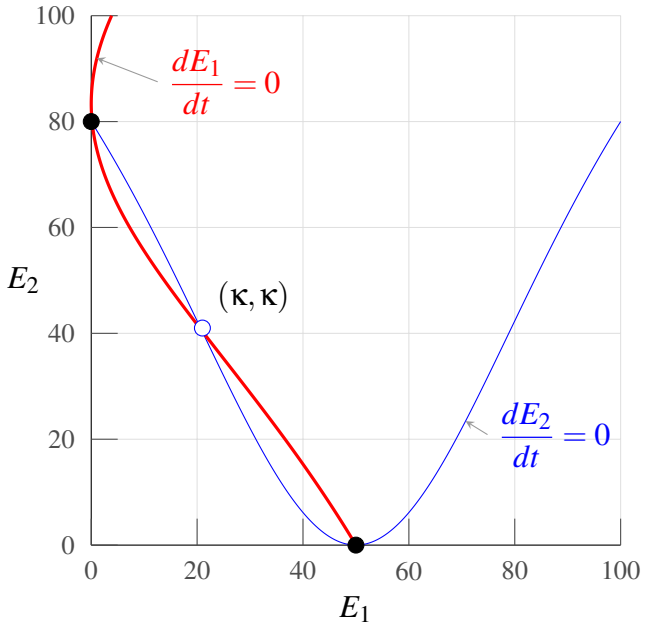


Figure 4.5.2: [Asymmetric connections] Isoclines for WTA network with asymmetric connections. The network is stable at either $(0, 80)$ or $(50, 0)$. This means that the ON activation of E_2 is larger than E_1. In addition, the domain of attraction of E_2 (not shown) will be larger than that of E_1.

This change in equilibrium state of the network is accompanied by a change in the domain of attraction of each equilibrium state. A domain of attraction for an equilibrium state can informally be defined as the set of all intial condition for which the trajectory approaches the equilibrium. In our case, when one of the connections becomes stronger than the other, then the node which receives the stronger inhibitory connection develops a smaller domain of attraction. This means that if equal size of external inputs are provided to the two nodes, then the node with the larger domain of attraction will dominate. This allows us to build an incremental learning mechanism that changes the connection strengths between the two nodes, making the node that has been activated more frequently have the larger domain of attraction.

The potential problem with this solution is that unless the strengths of connections are changed in a particular way, the system tends to develop equilibriums that do not completely turn one node OFF and the other ON. For example, in

figure 4.5.2, the peak activation as well as the external inputs of the two nodes had to be asymmetrical so that the isoclines intersect along the x and y axes. Therefore, this solution, even though plausible, is complex to implement.

- SENSITIVITY TO EXTERNAL INPUT. We have seen above how each node in a dynamical system is pulled by two forces: hysteresis and external input. The equilibrium of a dynamical system in which both nodes receive the same amount of external input will be governed by hysteresis. However, if the external input is biased towards one of the nodes, then the equilibrium will be governed by both hysteresis and external input. If the difference in external input is large, then this input can overcome the effect of hysteresis and the equilibrium will be completely determined by the external input. Therefore, one can think of a linear scale that shows the kind of forces that influence the equilibrium of the a dynamical system (figure 4.5.3).

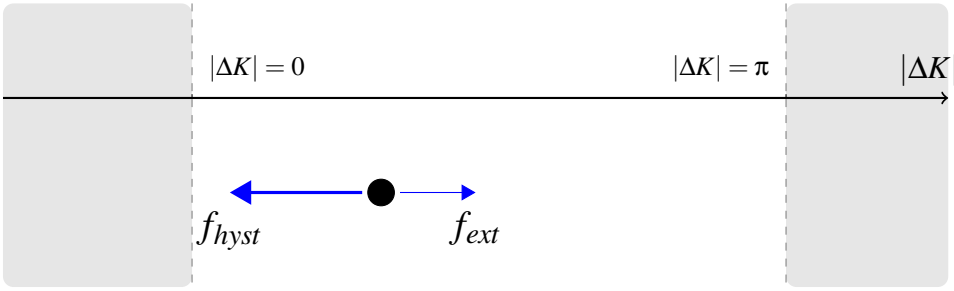


Figure 4.5.3: [Forces] Hysteresis and External input can be thought of as forces acting on the dynamical system.

The little circle below the line in figure 4.5.3 can be seen as the external input in the case of a particular trial. This external input is pulled towards the points $|\Delta K| = 0$ and $|\Delta K| = \pi$ by the two forces of hysteresis and external input. Now, consider a situation in which the external input available to each node is not fixed, but scaled by a factor, say ω_i for each node. A key point is that, in this case, the distance between the two points $|\Delta K| = 0$ and $|\Delta K| = \pi$ is not absolute, but dependent on these scaling factors. In case of a two-node network, if the two scaling factors are close to each other ($\omega_1 \approx \omega_2$) the distance between the two points will be larger than the case in which there is a large difference between ω_1 and ω_2. Therefore, a small difference in external input could lead to the node with larger input winning the competition, provided there is a large difference between the two scaling factors. These scaling factors, ω_i, can be seen as weights

attached to inputs, or as sensitivities of a network's inputs to external stimuli.

These input sensitivities can serve as the long-term record of the system's processing. Whenever a node gets activated (wins the competition, for WTA network), its sensitivity to the input can be increased by a small amount λ. This way, a node that gets activated greater number of times will be more sensitive to the input and consequently more likely to win the competition given the same external stimuli.

Each of the variables presented above provides a mechanism for encoding long-term learning. We choose the last one – 'Sensitivity to external input' – because it is the easiest to implement and most biologically plausible. Adjusting the input sensitivity is Hebbian learning in its most primitive form: when an input frequently contributes to firing of a particular neuron, then synapses from the input to the neuron should be strengthened.

The other two mechanisms have their limitations. Implementing long-term learning as adjustment to the strength of excitatory and inhibitory links suffers from computational complexity. As the network adjusts the strength of links between nodes the domain of attraction for each equilibrium, as well as the position of the equilibrium, will change. We need to make sure that the learning mechanism leads to stable solutions and it guarantees that repeated presentation of stimuli will lead to convergence of connection strengths to finite values. In the next chapter we will see that we can study such domains of attraction by defining a certain function called the Lyapunov function for the dynamical system. However, we avoid this theoretical complexity at the moment and use the simple mechanism of adjusting input sensitivity to investigate the effects of long-term learning on the behaviour of our system. We also avoid using the mechanism of adjusting the peak activation of a node as we saw that such a mechanism will show a sensitivity to noise and will be difficult to justify biologically.

Now, let us look at how we can implement learning using this mechanism in Model ③. In order to do this, first we need to formalise what we mean by input sensitivity and how we can adjust this sensitivity. Let us assume that the variable for recording this input sensitivity for each node is ω_i^t. The subscript i stands for the number of the node and the subscript t stands for time. We want the input sensitivity to accumulate over time. We want its current value to depend on its past values. We also want the input sensitivity to change if the node is activated. Therefore, the learning rule for change in

ω_i with each episode can be stated as:

$$\omega_i^t = \delta\, \omega_i^{t-1} + \lambda \tag{4.5.1}$$

where δ is a parameter that governs the effect of previous episodes on the current input sensitivity and λ is the amount of adjustment to input sensitivity as a result of the current episode.

The next thing we need to specify is how a node's input is affected by its input sensitivity. Let the external input to a node at time t be K_{ext}^t and let the input that reaches the node be K_{int}^t. We stated above that the input sensitivity can change the $|\Delta K|$ from an absolute value to a relative one. In other words, ω_i scales the external input: $K_{int}^t = \omega_i^t \times K_{ext}^t$. Alternatively, we can maintain this input sensitivity as an internal variable that is added to the external stimulus:

$$K_{int}^t = K_{ext}^t + \omega_i^t \tag{4.5.2}$$

Thus the input received by a node is the sum of the external input and an internal long-term learning variable. By examining figure 4.5.3, we can see that the larger the value of this internal variable, the more the input will be pushed towards $|\Delta K| = \pi$ and the more likely it will be that the external input will dominate hysteresis. Thus ω_i^t does indeed serve as the input sensitivity of a node.

Two unknown factors remain before we have completely defined the learning rule: δ, which is the effect of previous episodes on input sensitivity and λ, the effect of the current episode on input sensitivity. These factors can be, respectively, seen as the *decay rate* and the *learning rate* in long-term memory. First, let us look at the decay rate, δ.

§ 4.5.2.2. **Choosing the rate of decay δ.—** Any capacity-limited memory needs to undergo gradual forgetting in order to avoid losing all stored information or else losing the ability to form new memories (Sandberg, Lansner, Petersson, & Ekeberg, 2000). We have already seen one such mechanism of forgetting – adaptation in the activation of nodes – which leads to forgetting in the short-term memory. In long-term memory, we implement forgetting by building it explicitly in the learning rule 4.5.1. With each episode, the sensitivity of a node to input stimuli is adjusted based on the node's response to the stimuli. Thus the input sensitivity carries a record of a sequence of responses of the node and serves as long-term memory. This record, in turn, plays a role in calculating the new value of the input sensitivity. The learning rule is recursive

and with each step (episode), the previous input sensitivity is scaled by a factor δ[9], $0 \leqslant \delta \leqslant 1$.

What can be a suitable value for δ? If $\delta = 0$, then the input sensitivity changes with each episode. In this case, the system's memory is restricted to its immediately preceding episode. Therefore, the system can be represented as a finite state machine. At the other extreme, if $\delta = 1$, then the system shows no decay in memory. Each episode is accumulated in the memory and the system shows no recency effect – an episode in the distant past is as important as a recent one. Neither extreme is suitable for encoding structural repetition.

Szmrecsanyi (2006) showed that, in dialogue corpora, structural persistence does indeed decrease as the textual distance between prime and target increases. Furthermore, he compared the amount of decrease under two cases: immediately after the choice and a long time after the choice. He found a rapid decay in persistence immediately after a syntactic choice and a more gradual decay as time went on. Based on this evidence, he argued that the decline in persistence is best described by a logarithmic forgetting function. Other studies (e.g., Gries, 2005) have obtained similar results.

This behaviour of a rapid decay in memory immediately after an event and a slower decay as time goes on is not unique to structural priming and has a long tradition in the study of human memory. Researchers have proposed various mathematical functions that can describe how we forget memory traces. Wickelgren (1970) described the performance of subjects in a recognition experiment using an exponential curve. Rubin (1982) and J. R. Anderson and Schooler (1991) have described forgetting using a power function.

The recurrence of such functions in memory and other aspects of psychology prompted J. R. Anderson (1990) to put forward a theory of how such curves emerge as a result of interaction between a goal-oriented system and the natural environment. J. R. Anderson (1990) compared memory recall to nonhuman information retrieval systems such as the library borrowing system and computerised file systems. Information retrieval in these nonhuman systems can be predicted by using the history of these systems – i.e. the statistics of how these systems have been accessed in the past. In a similar manner, J. R. Anderson (1990) argued that recall in human memory can be predicted by using the history of how the memory system has been accessed in the past (and, of course, the recall cues). This statistical history of memory retrieval is what

[9]This factor, which is a constant, is a simplification that encompasses the rate of change in long-term memory. This change could, in turn, depend on a number of factors and the constant δ, in that case, could be replaced by a function $\delta, (\delta \in \mathbf{R} \rightarrow \mathbf{R})$.

J. R. Anderson (1990) called the memory's environment. He shows that the structure of this environment can be formally related to the probability of recall from memory. Crucially, if one assumes that the environment has a certain structure, then without making any additional assumptions about memory mechanisms, J. R. Anderson (1990) showed that memory decays via a power function. Thus, his analysis showed that decay curves such as the power function can arise out of the memory's statistical history of usage.

Both the exponential function and the power function possess the property of rapid decay immediately after encoding and a more gradual decay as time goes on. The two functions however differ in how quickly memory decays with exponential decay being faster than decay under a power function. Rubin (1982), J. R. Anderson (1990) and J. R. Anderson and Schooler (1991) have argued for power function to be used to describe recall and recognition in memory. However Szmrecsanyi (2006) and Gries (2005) have shown an exponential decay in structural priming with time. We will eschew from the debate between these two functions and because we are studying structural priming, we will adopt the exponential function to describe decay in our system. However, the mechanistic framework presented in this chapter does not depend upon the nature of the function and if a future experimental study shows compelling case for structural priming to follow the power function, we can revise our model to implement such a function.

In Equation 4.5.1, the input sensitivity ω becomes a fraction of itself after each episode. Therefore this function implements an exponential curve and we can control the rate of forgetting for this exponential curve by specifying δ as

$$\delta = \frac{1}{\exp(\tau_\delta)} \tag{4.5.3}$$

where τ_δ is a parameter that governs the speed at which the exponential changes. This equation satisfies our criterion of having a value of δ between the two extremes: $0 < \delta < 1$.

§ 4.5.2.3. **Rate of learning λ.—** Now that we have decided how to represent the amount of decay with each episode, let us turn our attention to the amount of learning, or increment in input sensitivity, λ. The key question regarding the amount of learning is whether it should be fixed for each episode, or dependent on properties of the episode.

From a behavioural perspective, we do not learn equally from all episodes. This might have to do with the amount (or kind) of information present in the episode. Surprising episodes have more information content and require greater amount of learning.

If the system's goal is to model the input stimuli, then it will have variable amount of learning for each episode.

On the other hand, from the system's perspective, each flow of information through the system leaves a memory trace in the system that corresponds to a fixed amount of learning. One could argue that variable amount of learning is actually a consequence of processes such as rehearsal, which take place only at the command of a higher-level module that performs a meta-analysis of the input information. So if the system's goal is to model memory in a subsystem then it should show a fixed amount of learning.

Thus the choice of whether learning should be fixed or variable depends on two contrasting perspectives of learning – each with its set of arguments. We need to base our choice of learning algorithm on how relevant are each of these opposing arguments to the study of structural priming. Let us consider each of these arguments in a bit more detail.

Consider, for example, a generative model that tries to reproduce a pattern of data points. This model might have some internal representation of the data, let us say a mixture of two Gaussians. The learning algorithm will adjust the parameters of the model – its mean, variance and mixing proportions – to form a best fit to the data. While some data points such as the ones close to the estimated mean of a Gaussian, will lead to a small amount of learning, an outlier will lead to larger adjustment to the estimated parameters. Therefore, this generative model will show a variable amount of learning.

Another example is an error-based learning model, such as Chang et al. (2006), which calculates the error by comparing the output of the model with a known target and makes adjustments based on the difference between the output and the target. Episodes in which the output (or prediction) of the model is close to the target, lead to a small adjustment or error-correction. While episodes in which the prediction is very different from the target lead to a larger error-correction. Thus the amount of learning in the model is, again, variable.

These two examples can be contrasted with the kind of model we are trying to develop in this chapter. In our model, long-term learning is a cumulative record of trailing activation from a sequence of episodes. The dynamical system treats each episode in the same way – with competition between nodes and the winning node achieving maximum activation. The system's input sensitivity needs to simply record the memory of the episode and unlike the two examples above, it is not concerned with finding an internal representation of the statistics of the environment.

There is no denying the fact that subjects might actually perform a variable amount of learning in different episodes. We noted above, for example, that surprising episodes contain greater amount of information and therefore might need larger amount of learning. But this variable rate of learning might actually be moderated by a different cognitive system, like the attentional system. Carlson and Dulany (1985), for example, show that subjects learn attended stimuli better than unattended or background stimuli.

In this study we do not intend to develop a holistic model of human cognition. In particular, Model ③ does not look at the effect of such attentional processes – instead, it teases apart a memory module and looks at the duration of priming and lexical boost in such a memory module. For this reason, we assume a fixed amount of learning, λ, with each episode. Every time a node wins the competition, its input sensitivity increases by a constant $|\Lambda|$. On the other hand, every time the node loses the competition the sensitivity decreases by the same amount $|\Lambda|$:

$$\lambda = \begin{cases} +|\Lambda| & if \quad ON \\ -|\Lambda| & if \quad OFF \end{cases} \tag{4.5.4}$$

In this algorithm, we choose to both increase and decrease the input sensitivity of the learning parameter because it prevents saturation. The value of $|\Lambda|$ is a free parameter that governs how quickly or slowly a system learns with each episode.

Our choice of a fixed learning rate allows us to highlight another feature of the current model that distinguishes it from some other models. One way to distinguish our model from an account like Chang et al. (2006) is to say that the current model is based on unsupervised learning while their model is based on supervised learning. While this distinction is true, we would like to argue that the difference between our models is not limited to the usage of supervised versus unsupervised learning algorithm. We have used unsupervised learning to encode traces of memory that are created in the system as a result of information flow. However, unsupervised learning can also be used by generative and recognition models to reproduce input stimuli. These models try to look for a set of causes underlying input data. The Gaussian mixture model that we discussed above, for example, can learn through an unsupervised algorithm, but its goal is to reproduce the input data. Looking for such a set of underlying causes is a goal of language acquisition. However, our goal is not to model language acquisition. Instead, we are modelling linguistic memory. The key difference is that we do not assume that this memory is used to look for a set of causes in the input stimuli. Rather, it is a by-product of the flow of information through the system. Thus, not only is the learning algorithm in our model unsupervised – i.e. learns without a teacher – but it

is also 'indifferent' – i.e. it does not try to find underlying causes in input stimuli. In other words, the function of learning in our system is not to *acquire*, but to *remember*.

§ 4.5.3 Simulation & Results

§ 4.5.3.1. **Experiment Design.—** The simulations on Model ③ tested the long-term persistence of structural priming and the lexical enhancement of this priming. The phrase 'long-term' is used in this context to specifically refer to testing conditions where prime and target were separated by *interfering* episodes. The term 'interfering' distinguishes these set of simulations from those conducted on Model ②. There, the prime and target trials were separated by an adaptation period. That adaptation period led to a decay in memory. However, the constructs that were activated during the priming trial were not re-activated or reset between the prime and target trials. We use the term non-interference to refer to this condition. This non-interfering period of decay was simulated through the loss of activation in the WTA and STM nodes. It is possible that the system receives filler trials during this adaptation period, but these trials do not invoke the same constructs (combinatorial nodes) as those in the prime and target and hence do not interfere with their activations.

In contrast, during the simulations in this section, the activity of a construct might be pushed towards ON or OFF states between a prime and target trial. In other words, these simulations test the affect of a sequence of priming trial on a target trial. Each priming trial simulates comprehension and therefore the winning nodes in each WTA layer are completely determined by external stimuli. A pictorial representation of possible simulation is shown in Figure 4.5.4. This simulation consists of two priming episodes followed by an adaptation period, which is then followed by a target trial. At the end of the target trial, the simulation adjusts the input sensitivity of all the nodes. The reader would recall that the input sensitivity stores the long-term memory of the system. The syntactic choice made during the target trial is governed by both the input from STM module and the sensitivity of the combinatorial nodes to input stimuli.

In order to make direct comparisons with behavioural data, we tested the model under same experimental conditions as Kaschak and Borreggine (2008). Each of their experiments is divided into two phases: a training phase, and a testing phase[10] (Figure 4.5.5). Subjects are trained for a particular grammatical construction during the

[10]The testing phase does not imply that the model has stopped learning – it performs learning during both phases, but is tested on target sentences only during testing phase.

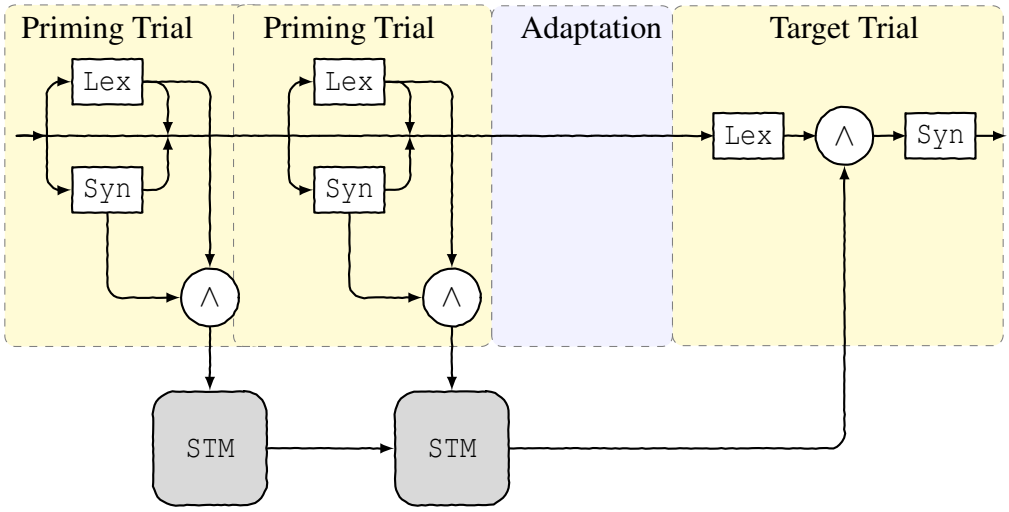


Figure 4.5.4: [Experiment Design] Model ③ tests priming over a sequence of episodes. This diagram illustrates a sequence of two priming trials followed by a target trial. In practice, a longer sequence was used for testing (see figure 4.5.5).

training phase by being coerced to produce it. These are shown as the 'prime' trials in Figure 4.5.5. After 10 such priming trials, subjects receive 6 prime-target pairs during the testing phase. Simulating each prime-target pair is exactly same as the simulation for Model ② (Figure 4.4.4). The only difference is that the external input to combinatorial nodes during the target trial is moderated by the input sensitivity of the node. This input sensitivity has been adjusted as a result of the sequence of priming trials during the training phase. As a result of this adjustment, we expect priming to be influenced by both short-term dynamics and long-term learning. The results can then be compared with the behavioural data observed by Kaschak and Borreggine (2008).

Kaschak and Borreggine (2008) found a cumulative effect of priming – i.e. the training phase influences the amount of priming during testing phase. However, they found that this long-term influence is not mediated by the choice of verbs. Based on this result, they have argued that structural priming is long-lasting while the lexical influence on structural choice is short-lived (see Section 2.3 for details). Our model allows us to test this hypothesis. We saw in the previous section (Equation 4.5.3 on page 146), that we can control the rate of learning and forgetting for each of the WTA layers and for the binding nodes. We also know that binding nodes provide a causal link from the lexical layer to the syntax layer. By varying the rate of long-term forgetting in binding nodes, we can vary the duration of lexical influence on syntactic choice. Similarly we can vary the duration of long-term priming by varying the rate

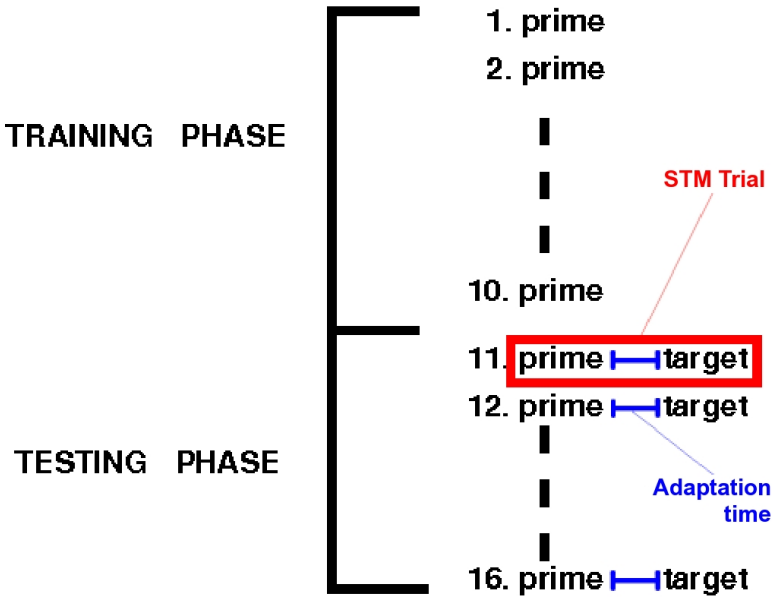


Figure 4.5.5: Experimental design for testing long-term priming and lexical influence.

of forgetting in WTA layer. In this section, we test our model when the rate of forgetting is similar for structural memory and the binding nodes. Assuming a similar rate of decay in the two kinds of representations is in contrast to the conclusion made by Kaschak and Borreggine (2008). If our model, based on a different set of assumptions (i.e. same rate of decay in syntactic layer and binding nodes), gives the same results as Kaschak and Borreggine (2008), then it would challenge this conclusion.

§ 4.5.3.2. **Input file.—** Before we proceed with the testing, we need to configure our model to receive the input stimuli required by the experimental design. Models ① and ② picked the input stimuli randomly for each priming trial. Our goal was to establish whether the combinatorial choices are repeated between a single prime and target. However, in this section, we are interested in investigating whether the combinatorial choices are primed by a sequence of trials and whether this priming is verb-specific – i.e. if we change the set of verbs between training and testing phases, does the amount of priming remain the same? To test these issues, the input stimulus needs to be rigidly defined for each priming trial.

An input file was constructed that specifies the choices for each priming trial. At the beginning of each trial, the simulation reads this input file and calculates the external input for each node based on the entry in the file. Unlike the previous simulations, this means that the model cannot choose a lexical node at random. It is not sufficient

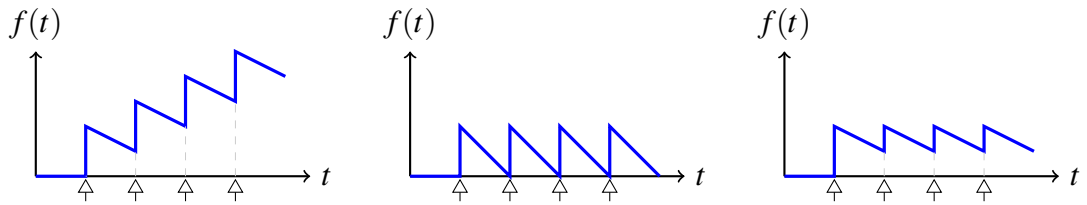
to have an equal probability of lexical selection for all the nodes. Rather, the external input must specify which node is selected. This chosen node gets a larger input than all others ($\mu \gg \theta$), thereby selecting the node.

This input file also allows us to run our entire simulation as a batch process. External input is specified for all sixteen episodes, for each of the eighty subjects. To provide a particular pattern of primes during the training phase, only this input file needs to be changed. The model calculates the appropriate inputs for each node after reading this file. Part of a sample input file is shown in appendix B.

§ 4.5.3.3. **Measuring Priming.—** So far, measuring priming has been simple. We count the number of repetitions between priming and target trials and compare these repetitions under priming and non-priming conditions (section 4.3.3.2). But now we want to calculate the amount of cumulative priming from a sequence of primes to a target. Therefore, we need a new measure.

Over a sequence of primes, what is important is the statistical pattern of the primes. Do all the priming trials favour one structure over the other? If not, is one structure more frequent than the other? These statistical properties of primes are likely to get encoded in our memory. If priming relies on incremental learning, it will accumulate the statistical properties of linguistic processing. If on the other hand, it is non-incremental then these long-term statistical properties will not be recorded. Priming might be non-incremental due to various possibilities. One of these possibilities is that it might be short-term. Figure 4.5.6 illustrates this idea. It shows three functions, with x axis showing time and y axis showing output of the function. The function on the left decays slowly and receives a boost whenever it receives an input (shown as an arrow along the x axis). It accumulates each input as the output of the function. The one in the middle decays very fast. As a result, its not able to accumulate the inputs. Therefore, it is not incremental. The function on the right is also not incremental, even though it is slowly decaying. This function is reset to a fixed value whenever it receives an input.

If priming is long-term and relies on principles of incremental learning then it will encode some statistical properties of the pattern of primes. Of course, priming might be non-incremental or only short-term. But we have a lot of evidence against these possibilities (Bock & Griffin, 2000; Chang et al., 2006). One way to measure the long-term effect of priming is to pick a statistical property of the primes (such as the ratio of POs to DOs) and see if this property affects priming. This is the method used in Kaschak and Borreggine (2008) and since we are replicating their experiment,



(a) Incremental (b) Fast forgetting (c) Fixed response

Figure 4.5.6: [Types of learning] Function $f(t)$ records events related to the system. These events (shown as arrows along the x axis) can be activation of node, or arrival of an input, etc. In our model, the function $f(t)$ can be seen as input sensitivity. (a) shows a desirable incremental function that accumulates a record of events; (b) shows a function that does not accumulate a record because it forgets very quickly and (c) shows another function that does not accumulate the record because each event resets its value to a constant.

we will follow this measurement. Of course, it does not follow that if the statistical property does not affect priming, then priming is non-incremental. We will discuss this possibility in section 4.6.

To quantify the pattern of priming, Kaschak and Borreggine (2008) compared the amount of priming for two patterns of priming during the training phase. The first pattern, called *Equal condition*, consisted of an equal number of primes of each grammatical construction. Since we are looking at alternative dative constructions, this condition presented subjects with the same number of PO and DO constructions. The second pattern, called *Unequal condition*, consisted of all primes having the same grammatical construction. That is, all ten prime trials during training phase used the same dative structure.

Kaschak and Borreggine (2008) put forward the following hypothesis. If priming accumulates over several trials, then the Unequal condition will bias the subjects towards a particular construction. This bias should interfere with the priming between a prime-target pair during the testing phase. On the other hand, the Equal condition will not bias the subjects towards any particular construction. Hence it should not interfere with priming between a prime-target pair. Thus, a difference between priming under Equal and Unequal conditions should indicate a long-term effect of priming. Furthermore, the value of this difference can be used as a measurement of the amount of priming from training to testing phase.

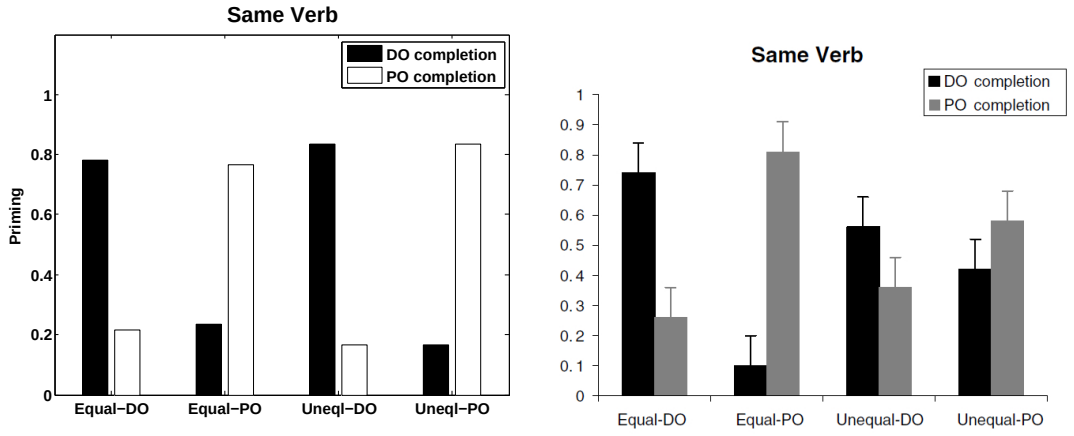
Lastly, we need a method for measuring lexical enhancement of structural priming over a set of trials. Again, this measurement was simple in the short-term where we could compare priming in cases there was lexical repetition ($Priming_{rep}$) to priming in general (*Priming*). However, over a series of trials, some trials might be using the same verb while others might be using a different verb. Therefore, we cannot use the same measure for lexical boost. Instead, Kaschak and Borreggine (2008) measure whether the priming is verb-specific or not. That is, they measure the difference in priming between Equal and Unequal conditions when the same set of verbs are used in training and testing conditions. This difference is then compared to the corresponding difference when a different set of verbs are used in the two phases. These conditions have been called the *Same Verb condition* and the *Different Verb condition*, respectively. If the pattern of priming is different for Same and Different Verb conditions, then priming is clearly verb-specific – i.e. there is long-term lexical influence on structural priming. Otherwise, Kaschak and Borreggine (2008) conclude that lexical influence is constrained to prime-target trials and does not have a cumulative effect.

§ 4.5.3.4. **Results.—** The model runs this simulation for eighty different subjects. Each subject receives a set of sixteen episodes, as shown in the experiment design above. The priming trial starts from the rest state and the target trial starts from the final state of the prime trial. The equilibrium state during the priming episode is completely determined by the external input. This is very much like the simulations on model ① and ②, with one difference. When we ran the priming trials for the previous two models, we randomly determined which syntax (and lexical) node wins the competition. We did not care which node won the competition; we only cared about whether the same node won the competition between prime and target trials. During the current simulation, we intend to replicate the experiment design for Kaschak and Borreggine (2008). In this design, the priming trial is a comprehension trial where the lexical and syntactic decisions are *not* made by the subject. Instead these decisions are already provided in an input file, as we saw above. Therefore, the external stimulus for the priming trial comes from an input file.

The target trials proceed in the same manner as in the previous models. The external stimulus is randomly obtained from the probability distribution $p(|\Delta K_{test}|)$ which is modelled as a Gaussian with mean μ and standard deviation σ. We saw in section 4.3.2.1 (page 113) how the parameters of this distribution can be determined using the balance between hysteresis and external input (π) and the bifurcation parameter of

the dynamical system (θ). In order to test this model, we developed a graphical user interface that automates the calculation of μ and σ based on the specification of an overall level of hysteresis by the experimenter. This GUI (shown in appendix A) also allows the experimenter to specify other parameters of the model such as the adaptation time between prime and target trials in an episode and the long-term learning rate, Λ, between episodes (Equation 4.5.4).

Our first simulation tested whether the model shows structural priming for the experimental setup used by Kaschak and Borreggine (2008). We were interested in checking whether the model showed short-term priming – i.e. priming from a comprehension to a production trial within an episode. Therefore, we decreased the long-term learning parameter, Λ, to zero. We also assumed an intermediate value of $\pi = 0.4$ – i.e. there is a 40% chance that hysteresis will dominate external input. Lastly, we set the adaptation time to be 1000ms. The reader would recall from the results of the previous model (table 4.1) that at this value of adaptation time, the model still showed both structural priming and lexical boost. The results of the simulation are shown in figure 4.5.7.



(a) Results from simulating model ③

(b) Results from Kaschak and Borreggine (2008)

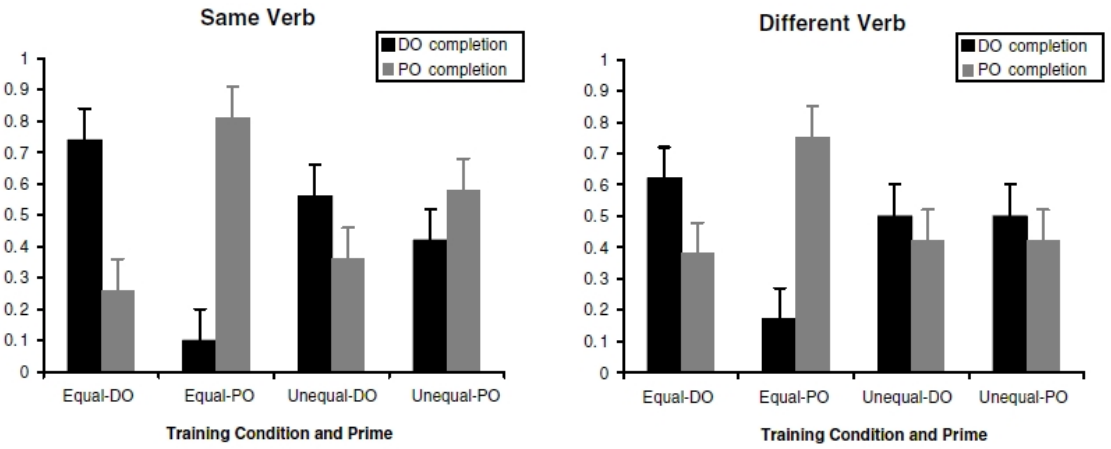
Figure 4.5.7: A comparison of results from simulating model ② and the experimental study conducted by Kaschak and Borreggine (2008) (Same Verb condition). Each graph shows the proportion of PO and DO completions in the production trial for the four priming conditions. Results for the simulation generated with the long-term learning turned off.

These results are presented in the same structure as the results given by Kaschak and Borreggine (2008). There can be two types of priming conditions in the training phase: *Equal* and *Unequal*, and two types of primes during the testing phase: PO or

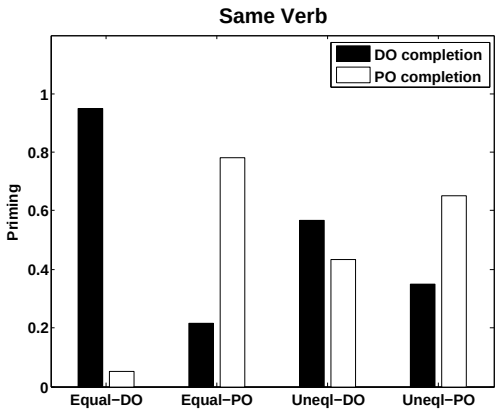
DO. This leads to four (2×2) types of prime conditions, which are shown along the x axis. The y axis is the dependent variable and measures the proportion of PO and DO sentence completions, in response to each priming condition. From this figure we can see that the DO prime led to a larger proportion of DO completions, as compared to PO completions. Similarly the PO prime led to a larger proportion of PO completions. Thus each syntactic structure led to an enhancement for the subsequent selection of that structure – i.e. the model showed structural priming. It can also be observed from the figure that the model showed similar amount of structural priming for the Equal and Unequal conditions. Because the Equal and Unequal conditions manipulated the sequence of primes during the training phase, these results imply that the model showed no effect of the training phase on structural priming – i.e. it showed no long-term priming. Indeed, we were not expecting the model to show such a long-term priming because we had set the long-term learning parameter Λ to zero.

In our next simulation, we wanted to test whether the model showed long-term priming and also whether it showed a long-term persistence of lexical boost. As we discussed above, we cannot measure lexical boost directly over a series of trials. Instead, following Kaschak and Borreggine (2008), we measure the verb-specificity of structural priming. For doing this, we needed to test the model under the *Same Verb condition* and the *Different Verb condition*. We put half of the subjects into each condition – i.e. forty subjects were trained and tested under the Same Verb condition and forty under the Different Verb condition.

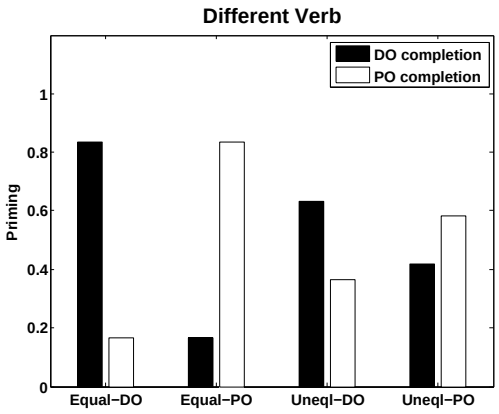
Crucially, we set the learning rate Λ to a positive constant 0.8 (the parameter is normalised between 0 and 1). This means that the model should now start accumulating the effects of each priming trial during the training phase. Activating the long-term learning also requires the specification of a second parameter, the rate of forgetting δ (Equation 4.5.1 on page 4.5.1). We set δ to a positive constant as well by setting the time constant for change in δ to 0.3 – i.e. $\tau_\delta = 0.3$ (Equation 4.5.3). Other parameters such as the adaptation time, μ and σ remained same as the previous simulation. By setting the parameters in this manner, we made the model capable of showing both a short-term structural priming and a long-term accumulation of priming as a change to the input sensitivity of nodes. Lastly, the model assumes a similar rate of decay in the syntactic layer and the binding nodes – setting the time constant for decay in input sensitivity of binding nodes to be 0.3 and the time constant for adaptation to be 4000ms for both the WTA and STM networks. Figure 4.5.8 shows the results of simulating the model under Same and Different verb conditions.



(a) Results from Kaschak and Borreggine (2008)



(b) Simulation results for Same Verb



(c) Simulation results for Different Verb

Figure 4.5.8: Simulation results for Experiment 1 from Kaschak and Borreggine (2008). The y-axis estimates the amount of priming by measuring the proportion of PO and DO completions.

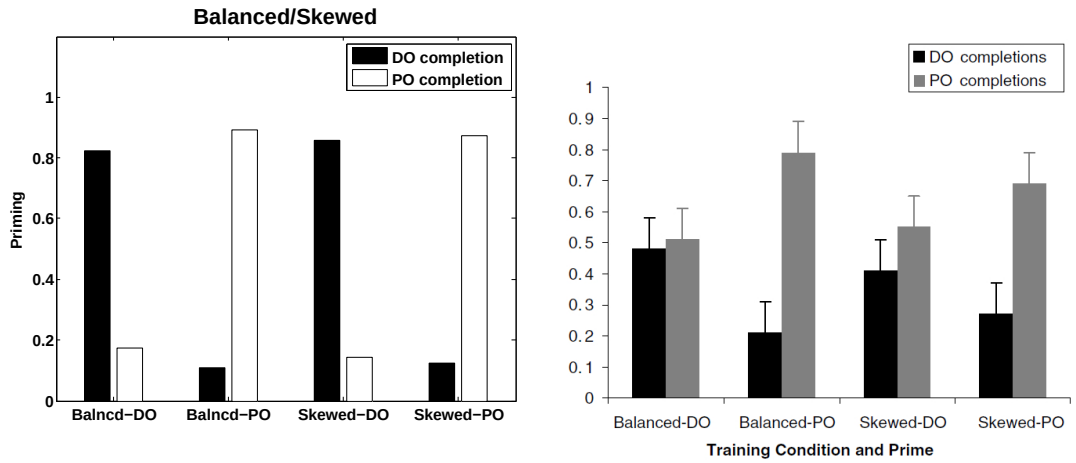
We can contrast these results to the results of our previous simulation (Figure 4.5.7). First, let us look at the priming shown under the Equal condition. Under this condition, just like the previous results, the model showed a larger proportion of PO completions after a PO prime and DO completions after a DO prime. However, the priming shown under the Unequal condition is in contrast to the previous results. For both the Same and Different verb conditions, the proportion of PO completions after a PO prime and DO completions after a DO prime are diminished in the Unequal condition. Since the Equal and Unequal conditions manipulated the sequence of trials during the training phase, any difference in these conditions must be due to the long-term learning incurred during that phase. This reduction in structural priming is the

same result that was found by Kaschak and Borreggine (2008) and therefore the model replicated their findings. Furthermore, both the Same and Different verb conditions showed a similar reduction in priming for the Unequal condition. Again, this is a result that was obtained by Kaschak and Borreggine (2008), and one which they used to justify the lack of influence of lexical factors on long-term structural priming. These results are surprising because the computational model assumed the same rate of decay for syntactic nodes and their lexical enhancement – an assumption that is in contrast to the conclusion made by Kaschak and Borreggine (2008) that lexical enhancement of priming decays quickly while syntactic abstractions persists in long-term memory.

While the previous simulation tried to detect lexical enhancement of structural priming by comparing priming for different frequency verbs (Equal versus Unequal condition), it is certainly possible that lexical influence does not show up in this statistic. Kaschak and Borreggine (2008) used another method to detect the long-term lexical enhancement of priming. This alternative method compares the amount of priming for verbs that are equally associated with both the syntactic constructions during training phase to the priming for verbs that are associated with only one construction. These two conditions have been named *Balanced condition* and *Skewed condition*, respectively. The hypothesis is that if lexical enhancement is long-lasting, then verbs that are biased towards one construction (Skewed condition) should interfere in the priming during testing phase, decreasing this priming; in contrast, unbiased verbs should show no interference. Thus, this experiment involves measuring the amount of priming under the two conditions and comparing the results.

In order to conduct this experiment, another input data set was constructed where half of the subjects were put into the *Balanced* condition and the other half into the *Skewed* condition. Again a set of eighty subjects were simulated with the same parameters as above. The results of the simulations are shown in figure 4.5.9.

It can be observed from this figure that the model showed similar amount of structural priming under the Balanced and Skewed training conditions. In other words, the amount of priming does not diminish under the Skewed condition, as compared to the Balanced condition – which is the same result that was found by Kaschak and Borreggine (2008) in their second experiment. Thus, the model replicated their results – a finding that is again counterintuitive since this computational model works under different assumptions from the model of linguistic memory that they are trying to support. We explore the reasons behind these results and the consequences for the study of linguistic processing in section 4.6.3.



(a) Results from simulating model ③ (b) Results from Kaschak and Borreggine (2008)

Figure 4.5.9: Simulation and experimental results for Experiment 2 from Kaschak and Borreggine (2008). This experiment compares priming under balanced and skewed conditions.

4.6 Discussion

Psycholinguistic experimentation and computational modelling share the goal of understanding the psychology of language, or in our case, the psychology of structural priming. However, these two different methods make different assumptions and look for the answers in contrasting places. Psycholinguistic experiments look at behavioural correlates of structural priming. These experiments usually relate it to other aspects of environmental stimuli. Experiments exist which show that structural priming exists in the absence of overlap at lexical, thematic role or sentence prosody level (Bock, 1989; Bock & Loebell, 1990). Others show that it is influenced by its lexical context (Pickering & Branigan, 1998), that it exists for both monologue and in dialogue (Branigan et al., 2000) and that it exists across modalities (Hartsuiker et al., 2008).

But human memory is cognitively embedded. Therefore priming, which is a form of memory, is related not only to other environmental stimuli, but also to other cognitive processes. Computational accounts of structural priming try to model this relation. The model presented in Chang et al. (2006), for example, relates structural priming to learning and prediction. We presented three models in the last section. These models relate structural priming to the cognitive concepts of arousal, competition, fatigue and input sensitivity in the cognitive system. Admittedly, the implementation of these concepts is fairly simplistic. However, these models are a first attempt at understanding how structural priming might be influenced by physical and computational properties

of the system that implements language processing.

In order to develop a computational representation of a cognitive process, each model makes a set of assumptions. These assumptions are manifested in either the architecture of the system, or in the value of its parameters. Based on these assumptions, the simulations allow us to make a series of inferences. In this section, we discuss the assumptions underlying three cognitive properties: arousal, fatigue and incremental learning. We discuss the validity of these assumptions, possible alternatives and the set of inferences that these assumptions allow us to make.

§ 4.6.1 Arousal

In order to explain the observation that subjects show a variable degree of structural priming under different conditions, we formulated the concept of arousal. This arousal, we proposed, is a global property of the system that controls the amount of priming shown by the system. In this sense, we argued, arousal captures the trade-off between automatic and strategic processes involved in linguistic processing. As arousal increases, the system becomes less automatic and strategic constraints dominate linguistic decisions. When arousal decreases, the system becomes more automatic and the process of priming dominates linguistic decisions. Thus, we argue that priming is variable because automaticity is variable and since automaticity is related to the state of a system's arousal, we can argue that the real reason for the variation of priming is the variation of this state of arousal of the system. This reasoning, however, only shifts the burden of the explanation from automaticity to arousal; it raises the question why arousal should be variable. Our answer to this question requires a look into the decision making process of a computational system.

Any computational system makes a decision under a set of constraints. This set encompasses a number of (local) considerations or premises, but does not include other, more general (non-local) premises. Decisions are made in a *closed world* and can always be revised to include more general and distally related constraints. The more constraints that are considered, the more general the decision will be. However, the decision-making process will be more time-consuming and effortful. Therefore, the computational system has to strike a balance between these local and non-local constraints.

Assuming that the cognitive system is a computational system, it faces the same conundrum. Making, say, a structural decision could involve considerations of seman-

tics, pragmatics, fluency, audience-design, etc. Ideally, the cognitive system should consider each of these constraints before making the decision. However, making each structural decision under all these constraints will introduce unnecessary load on the cognitive system.

The solution to the conundrum involves our definition of the term 'arousal'. The amount of arousal in the system can be seen as the size of the set of constraints under which a decision is made. If the arousal is low, the set of constraints is small and the decision is local. If the arousal is high, non-local constraints are considered in making the decision. Thus arousal provides a mechanism that allows the system to select whether the decision will be local or non-local. And in this sense, the term arousal is closely related to the cognitive processes of *detecting* a set of constraints and *orienting* towards these constraints. As Posner and Petersen (1990) point out, these activities are two of the three major functions of the attention system in the brain. Therefore arousal is closely related to the concept of attention.

Model ①, therefore, makes the connection between attention and language processing explicit. It connects the dynamical systems making syntactic and lexical choices to a module maintaining the level of attention. In agreement with this argument, Horton and Keysar (1996) found that interlocutors took the point of view of of their listeners into account only when sufficient cognitive resources were available. When participants had to produce utterances rapidly, they frequently included information in their utterances that was not present in a shared context with their interlocutors. Thus the cognitive system, much like our computational system, adjusts the range of constraints that it uses to make linguistic decisions based on the cognitive resources that are available to the system.

Crucially, this system does not use any associative links between the lexical and syntactic layers. The results from simulating this model, though predictable from the theory, show that lexical boost can arise out of an alternative mechanism than such associative links. This alternative mechanism predicts that linguistic decisions are made under two contrasting forces:

- Winner-take-all dynamics which lead to *hysteresis* – i.e., the linguistic choice made by the system depends on the history of the system. This is the automatic process, taking only local constraints (of the syntactic layer) into consideration and hence involving a low level of arousal.

- An external input to the two nodes. Since the nodes compete with each other,

in the absence of hysteresis, the winner is decided by the external input ΔK_{test}. This is the strategic process, taking non-local constraints into account. These constraints could come, for example, from considerations of audience-design and manifest themselves as increase in $|\Delta K_{test}|$.

A key assumption of the model is to relate the external input to the arousal in the system. A high level of arousal leads to larger difference in external input, making it the dominant force in decision-making. A lower level of arousal will lead to a small external input, making hysteresis the dominant force in decision making. As arousal is inversely related to automaticity and hysteresis to priming, we can infer that a high amount of automaticity will lead to high priming and a low amount of automaticity will lead to low priming.

The second major assumption, which allows the model to simulate lexical boost, is that the level of arousal is a *global* property of the system. Instead of assuming that each layer has its own level of arousal, the model assumes that both the layers will show the same amount of arousal. Specifically, if the lexical layer is showing a low level of arousal, then so will the syntactic layer. Since arousal is inversely related to repetition, through a backward-reasoning, we can conclude that repetition in the lexical layer will mean a low level of arousal and consequently a small difference in external input to the combinatorial nodes. And a small external input implies that hysteresis becomes the dominating force leading to a larger amount of priming. In short, repetition in lexical layer is associated with increased priming in syntactic layer. This is the lexical boost effect (Pickering & Branigan, 1998).

Support for the hypothesis that repetition is accompanied by low level of arousal comes from observation of neural response to repetition. It has been observed (Miller & Desimone, 1994) that repetition leads to a decrease in neural response to the stimuli. This phenomenon is known as “repetition suppression”. Henson, Shallice, and Dolan (2000) also observed the contrasting effect – an increase in neural response to unfamiliar data. One way to explain repetition suppression is that fewer neurons respond to a repeated stimuli (Grill-Spector, Henson, & Martin, 2006) – i.e. representations for this stimulus become sharper and more sparse. In the context of decision-making, this would mean an increase in focus or decrease in generalisation. In other words, the set of constraints governing the decision would become smaller in size, being restricted to the essential local constraints. So, in short, repetition is accompanied by localised decision-making, which was our definition of a decrease in arousal.

At the start of the section we suggested that the notion of arousal can explain why

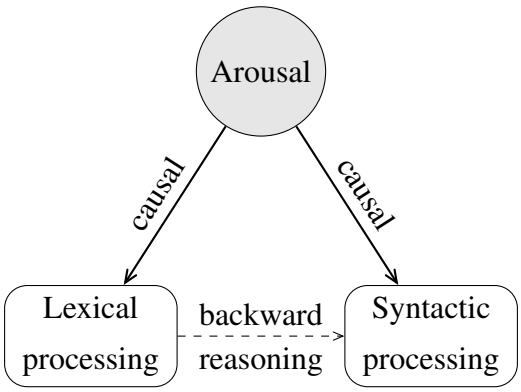


Figure 4.6.1: [Causal flow]

structural priming varies under different conditions. In previous studies, the variation in structural priming has been explained either in terms of the association between lexical and syntactic structures (Pickering & Branigan, 1998) or as a change in the degree of the automaticity of the system (Garrod & Pickering, 2007). Model ① suggests the alternative that structural priming can vary due to the range of constraints considered by the cognitive decision-making system. If the decision system considers only local constraints, then priming will be high. On the other hand, if the decision system considers a larger set of constraints, then the priming will reduce. According to this account, lexical repetition changes the amount of priming because the repetition of a word changes the amount of arousal in the system, which in turn changes the criteria under which the decision system makes the syntactic choice.

It is worth emphasising that Model ① does not have a causal flow of information between a lexical and syntactic layer. We have used backward reasoning to 'guess' that repetition in the lexical layer is a consequence of low arousal, which also leads to repetition in the syntactic layer. This model puts forward the idea that lexical boost arises as an epiphenomenon of a reduced level of external activation to modules of linguistic selection. The causal flow of information is from the global level of arousal to lexical and syntactic selection and not from lexical selection to syntactic selection (Figure 4.6.1).

Model ① is a theoretical account. Its truth can only be ascertained by checking the predictions of the model against experimental evidence. If the models assumptions are correct, then it makes two predictions which can be used to design future experiments:

1. PRIMING PREDICTION The model predicts that a low level of arousal will lead to greater priming. We have assumed that lexical repetition causes priming because

it leads to a state of low arousal. By this reasoning, other conditions that similarly lead to low arousal should also cause larger priming – i.e. lexical boost is only one is a set of different conditions that can lead to increased priming.

2. BOOST PREDICTION The partial correlation between structural (S) and lexical (L) repetition, given the level of arousal ($\mathcal{A}$) in the system, $\rho_{LS.\mathcal{A}}$, should be zero. That is, if we partial out the effect of arousal in the system, then lexical repetition should be uncorrelated to structural repetition. Again, we would need to manipulate the amount of arousal and record the amount of structural, lexical priming and their correlation.

§ 4.6.2 Adaptation

A central goal of the current study is to investigate the temporal properties of structural priming. In particular, we want to understand the mechanisms underlying decay in priming and in lexical enhancement of this priming. Model ② investigated one possible mechanism for such decay: fatigue. The idea comes from behaviour of neurons, which show an adaptation in its firing rate when exposed to a stimuli over a period. Instead of firing rates, our model remembers the stimuli in a stable state of a dynamical system. But just like neural adaptation, this system shows fatigue in maintaining its stability. That is, after some time, the stable state decays and the system loses its memory. Thus, if linguistic decision-making is implemented in such a system, then the temporal characteristics of this decision-making will be influenced by fatigue.

Model ② implements a process of adaptation through a decrease in activation of each node. We saw in the section 4.4.3, that such a system shows decrease in amount of priming as the period of adaptation increases (table 4.1 on page 136). Crucially, while the rate of priming decreases exponentially, the lexical influence on this priming decreases suddenly to zero after a certain period. This difference in the temporal properties of structural priming and lexical boost springs from the systems that implement these two kinds of memory.

Structural priming is implemented through hysteresis in a winner-take-all network. Each node in this network has inhibitory connections to all other nodes in the network. As the system approaches equilibrium, one node dominates all the others and wins the competition. This is the stable state of the network. But as the network progresses through the adaptation period, the winning node starts to show fatigue. Its level of activation starts to go down. At the same time, the activity of all the other nodes that it

had previously suppressed starts to increase. The network starts losing its memory.

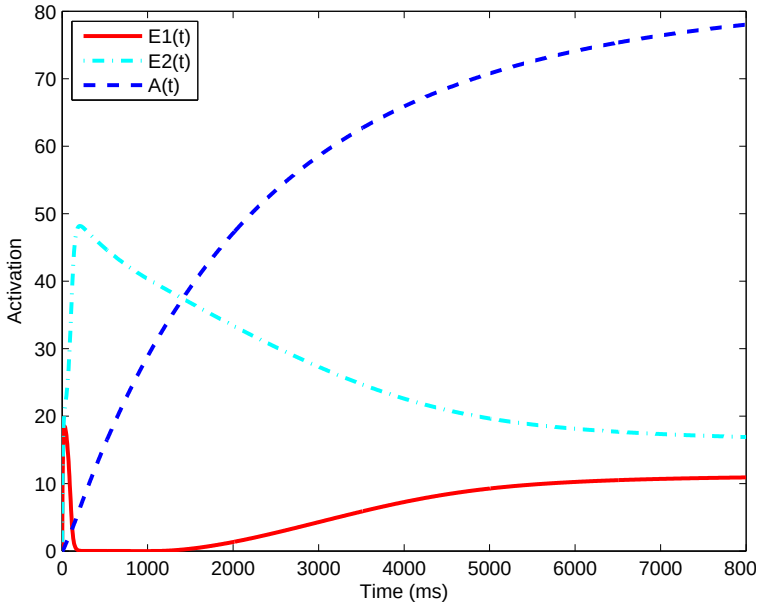


Figure 4.6.2: Adaptation in the winner-take-all dynamical system.

In Figure 4.6.2, we show the activity of the nodes in a WTA network consisting of two nodes. Time is plotted along the x axis and the activation is plotted against the y axis. It can be seen that as the adaptation period progresses, the activity of the two nodes approach each other. Since the memory of a WTA network lies in the difference in the activation of the two nodes, the network starts to lose its memory, which is apparent in the results in Table 4.1. A key observation is that (a) the activity of the two nodes approaches each other gradually and (b) the difference in activation asymptotically approaches a constant. As a result, structural priming in Model ② shows a gradual decrease, with saturation.

Lexical enhancement of structural priming comes through the input from the STM module. This module contains binding nodes retaining the conjunction between lexical and syntactic nodes. Each binding node is implemented as a mutually excitatory network. This network has two stable states – either both nodes are fully active (ON) for completely inactive (OFF). The binding node records a conjunction by turning both nodes to an ON state. Thus, the memory of the network resides in the stability of this ON state. As the adaptation period progresses, the activity of the nodes decreases and the stable equilibrium approaches an unstable saddle (see section 4.4.2 for details). The network shows a steady decrease in its memory while the stable equilibrium is ap-

proaching the saddle. At the bifurcation point, when the saddle meets the stable node, the system shows a complete loss of memory.

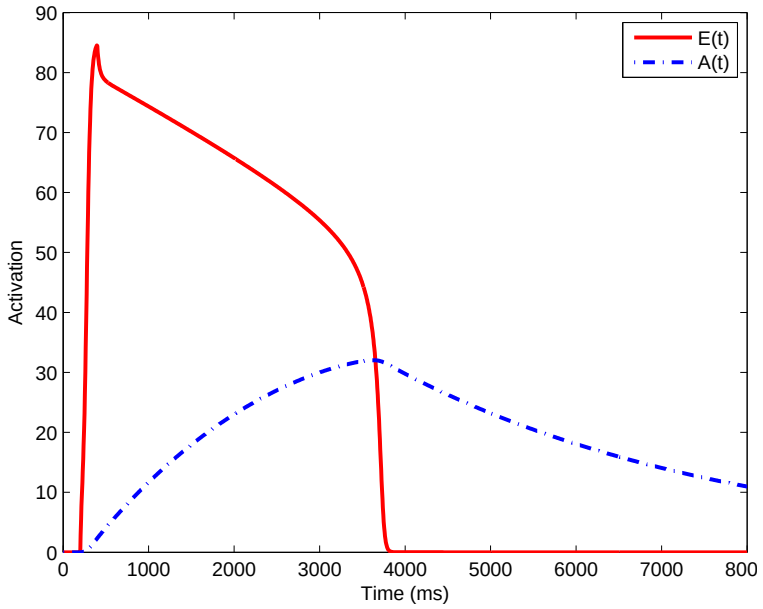


Figure 4.6.3: Adaptation in the mutualy excitatory binding nodes.

Figure 4.6.3 plots the activation of a typical binding node as the adaptation period progresses. Since both the nodes have the same activation, the figure plots the activation of one of the nodes against time. We can see the node seems to show an almost linear decay in activation up to a certain point in time (around 4000ms in the model). Beyond this time, the node shows a sudden decrease to zero activation – i.e. a *catastrophic* decay in its memory. In contrast to the WTA network, the STM network does not seem to show saturation. Also, for the same time constants, the STM memory seems to be short-lived, by comparison. This difference in the two kinds of memory is manifested in the results of model ② where lexical boost seems to suddenly decay to zero around 4000ms. It is no coincidence that the loss of stability in STM happens at the same time as the decay in lexical boost. The model's architecture constrains the lexical influence on structural decisions to act via the short-term memory and its decay inevitably leads to a decay in any lexical enhancement of structural priming.

Model ② gives a mechanistic explanation for the difference in temporal properties of structural priming and lexical enhancement of this priming. We can compare these results with the findings of Hartsuiker et al. (2008) who found that structural priming persists over large lags (up to six filler items), while lexical enhancement of this

priming decays quickly (see Chapter 2 for details of experiment). The results from simulating Model ② (table 4.1) mirror this effect. In their experiment, Hartsuiker et al. (2008) change the lag between prime and target trial by inserting filler trials. In our simulation, instead, we vary the length of adaptation period. Since the filler trials do not interfere with the syntactic or lexical constructs used in the priming trials, varying adaptation time achieves the same goal as varying the number of fillers.

Model ②'s explanation of the difference in structural priming and lexical boost relies on a crucial assumption in the architecture of the model: priming and lexical boost rely on different kinds of dynamical systems. Structural decisions are implemented through a WTA network where each node competes with all others. The lexical context of these structural decisions are maintained in a STM network, where nodes do not compete, but help each other stay active. One system relies on mutual inhibition while the other relies on mutual excitation.

This assumption makes sense. Activating a structural node is equivalent to making a decision about the structure of sentence. Only one structure can be used, so a choice needs to be made. For this reason, the decision-making network is competitive and uses mutual inhibition. Remembering the lexical context of a node, on the other hand, is not a decision-making process – it is a associative process. The binding node retains this association and nodes are mutually excitatory so that they can retain their activation in the absence of an external stimuli.

The validity of the model's assumptions can be checked against its predictions. Model ② asserts that syntactic decision-making and its lexical enhancement rely on two different cognitive systems. If this is true, then it should be possible to find a double dissociation between syntactic decision-making and influence of lexical context on these decisions. The short-term memory module is a domain-general cognitive system. It can be used to retain associations between linguistic constructs, but also associations within other modalities, such as vision (which has to solve the binding problem in object recognition) or cross-modality associations such as associating visual cues to co-occurring auditory signals. On the other hand, the decision-making module is domain-specific to linguistic processing. If the two types of networks have mutually exclusive existence in the brain, then it should be possible that one system gets impaired while the other is still functioning. Therefore, it should be possible that an individual has short-term memory impairment, but shows structural priming and equally, an individual who has an impairment with syntactic processing should show a larger lexical boost than control subjects.

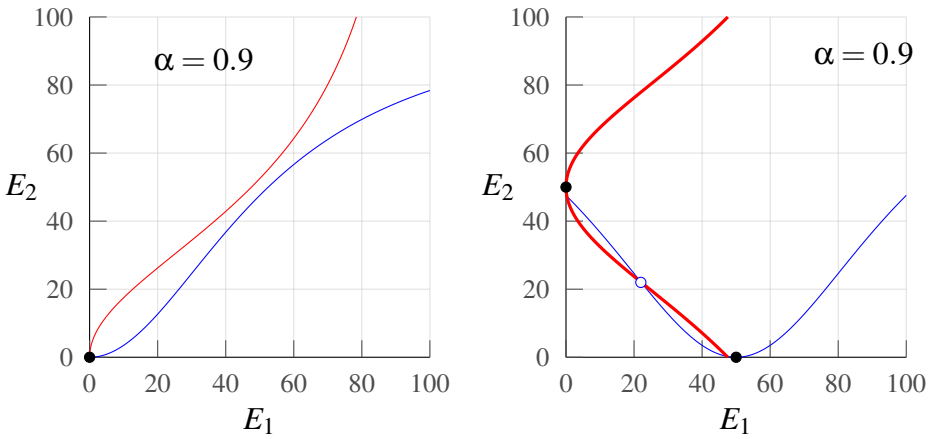
One more assumption lies at the heart of the difference between adaptation in WTA and STM networks. This assumption concerns the parameter α, the saturation constant for the adaptation variable in equations 4.4.2 and 4.4.3. When the adaptation period begins, both the WTA network and the STM network have a bistable solution. As the adaptation period progresses, the STM network crosses a bifurcation point and shows a qualitative change in its behaviour, becoming a monostable system. On the other hand, the WTA network never crosses the bifurcation point, remaining a bistable system. It is because of this difference in the trajectories of the dynamical systems that one shows a catastrophic decay in memory while the other shows an exponential decay with saturation. But this difference in behaviour is not guaranteed in all cases – it depends on the value of the parameter α.

Let us look at the function of the parameter α in greater detail. We saw in Section 4.4.2 that α governs the equilibrium value of the adaptation variable, A_i, expressed as a fraction of the activation of the nodes. That is, as the system approaches equilibrium, $A_i \rightarrow \alpha E_i$ (or, $A_i \rightarrow \alpha \sum_j E_j$ for Equation 4.4.3). The larger the value of α, the greater will be the value of the adaptation variable A_i. Since the variable A_i occurs in the denominator of the Naka-Rushton function ($\hbar$ is adjusted to $\hbar + A_i$ during adaptation), an increase in α, in turn, leads to a larger decrease in the activation E_i of a stable node. Thus α really governs how much the system will move from its stable equilibrium.

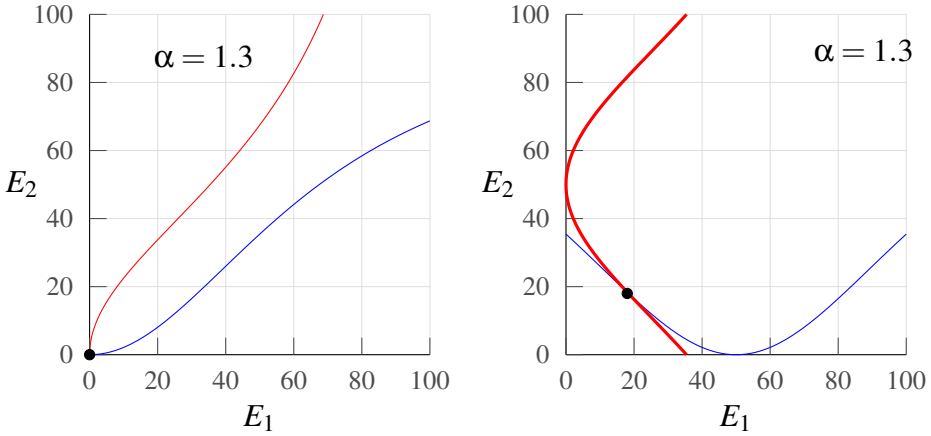
What is interesting from our perspective is that for certain values of α, the STM system will move enough from the stable equilibrium to be pushed over a bifurcation point while the WTA system will still be in a bistable state. This can be understood by looking at the change in phase plane of the two systems as the system goes through the adaptation period. Figure 4.6.4 plots the isoclines $\frac{dE_i}{dt} = 0$ for the two systems. For $\alpha = 0.9$, we can see that the STM system moves from bistability to monostability around $4000ms$[11] while the WTA system seems to remain in the bistable state even when A_i saturates (figure 4.6.4(a)). When we change this value of α to 1.3, both the STM and WTA systems move from monostability to bistability even though the STM system does seem to approach its bifurcation point more quickly (Figure 4.6.4(b)). Whether or not the WTA system crosses bifurcation seems to be contingent on the value of α. The results shown in Table 4.1 are obtained for $\alpha = 0.9$.

The theoretical constraint on α comes from the constraint on A_i, the adaptation variable. In equations 4.4.2 and 4.4.3, A_i changes the value of $\hbar$, the semi-saturation

[11] All other values of the parameters are as discussed in Table 4.1.



(a) Isoclines for $\alpha = 0.9$; A_i has reached saturation.



(b) Isoclines for $\alpha = 1.3$; A_i has reached saturation.

Figure 4.6.4: The same value of α could lead to different qualitative behaviours for STM and WTA system. When $\alpha = 0.9$, the STM system is monostable while WTA system is bistable (points of stability are shown as black dots).

constant. Recall that $\hbar$ is the value of input stimuli at which the response (of the node) reaches half of its maximum value (Figure 4.2.6). As the value of $\hbar$ increases, the S-shaped curve moves toward the right. The larger the value of A_i, the farther the S-shaped curve moves to the right. But there is a limit. This limit comes from the range of input stimlus. If the semi-saturation constant, $\hbar$, is larger than the input stimulus then the curve loses its S shape, no matter what the value of the exponent N. This is not functionally acceptable since the node never achieves its peak activation value. Therefore, we can limit the maximum value of A_i to the maximum input stimulus. The excitatory and inhibitory connections in STM and WTA networks mean that this input stimulus is, in turn, related to the output activation, E_i. So, as a conservative constraint,

we limit the maximum value of A_i to the output activation E_i (or to $\sum_j E_j$ for the WTA system). Since, at equlibrium, $A_i = \alpha E_i$, we obtain the constraint $\alpha < 1$.

As Figure 4.6.4 illustrates, under the constraint $\alpha < 1$ we can indeed obtain a difference in the qualitative behaviour of STM and WTA networks at saturation. Thus, we can observe that the difference we see in the temporal behaviour of priming and lexical boost might be due to the difference in the way these two phenomena are implemented in the system. On the one hand priming relies on WTA networks that show bistability at saturation. In contrast, lexical boost is implemented through the STM networks which degrade to monostability at saturation. These two behaviours are obtained for the same adaptation variable α.

§ 4.6.3 Incremental learning (and forgetting)

In the previous section, we saw how fatigue in the cognitive system might have contrasting effects on priming and lexical boost. In this section we look at another cognitive phenomenon: incremental learning, which is required when the system records information over a series of trials. The choice of the learning rule and the rate at which the system learns changes the behaviour of the system.

Model ③ builds in incremental learning and is able to record structural decisions over a sequence of trials. We saw that it is able to replicate experimental results on priming and lexical enhancement of this priming over a short period of time (e.g., Pickering & Branigan, 1998) and over a sequence of episodes (e.g., Kaschak & Borreggine, 2008). We chose the learning rule to make *fixed* adjustments to the system at the end of each episode. We also chose the rate of learning (and forgetting) in structural layer to be the same as that in the STM module. Thus the model had same rate of decay in the processes of syntactic decision and the channels through which lexemes influence this syntactic decision.

The results of simulating Model ③ for a sequence of priming episodes (figures 4.5.8 and 4.5.9) presented a paradox. The model was simulated on an input data similar to the experiments in Kaschak and Borreggine (2008). The results from these experiments showed no lexical enhancement of priming over a sequence of episodes. Our simulations replicated these results, but internally the model did not have a faster rate of decay in the lexical channel. This was surprising since the experimental design in Kaschak and Borreggine (2008) was supposed to detect any lexical influence on structural decisions. Because the simulations replicate these results, it suggests that lexical

influence shows a faster rate of decay than structural memory. According to the internal values of the decay parameters, we know that this is not the case. We would like to understand how the internal parameters for the rate of decay manifest in the behaviour of the model. Therefore, in this section we vary these rates and observe how this variation influences the behaviour of the model. We find that for certain values of these parameters, the null results from Kaschak and Borreggine (2008) do not necessarily imply a lack of long-term lexical influence on structural decisions.

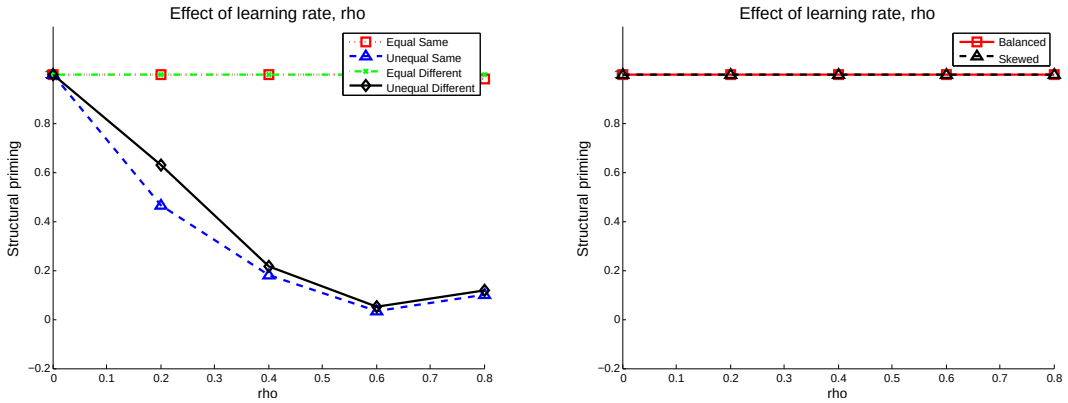
§ 4.6.3.1. **Learning rate in WTA layers, λ.—** The learning rate in WTA layers governs the amount of adjustment that is made to the input following each episode. Small learning rates help to make the system generalise when its trying to discover hidden variables in its environment. But small learning rates also mean that the impact of each individual episode is small. Large learning rates, on the other hand, mean that the system learns quickly, but also that the system becomes over-sensitive to recent episodes, thereby showing poor generalisation.

But the goal of the current study is different. We are considering adult speakers who have already acquired a working knowledge of language. Generalisation takes a back seat, as learning here is geared towards maintaining an active knowledge of dependencies within a sentence or within a discourse. A small learning rate would mean that the episode has not been recorded properly and that the increase in sensitivity of the node is small. This small increase in sensitivity might not be good enough to overcome the noise in the system. Hence the learning rate in our system is a direct reflection of how well an episode is recorded. Episodes with large learning rate will persist, while those with small rate will be forgotten quickly. Thus, we can control lexical and structural persistence in our system by varying the learning rate of each layer.

To check how this parameter affects structural repetition in the two experiments above, we repeated the above experiments under different values of learning rate, λ. These results are shown in figure 4.6.5, which plots the amount of priming[12] against the (normalised[13]) value of λ. In order to isolate the effect of λ, other variables were either set to zero, or to a default value. Learning in binding nodes, for example, was set to zero. This means that the binding nodes do not have any long-term learning. Adaptation time (i.e. the time between presenting a prime and a target) is set to 1000ms.

[12]The overall amount of priming is calculated as the average of the deviation from random responses (no priming) for the PO and DO prime conditions (Equation 4.3.6).

[13]Values of all variables are normalised between 0 and 1.



(a) Experiment 1: Priming under *Same Verb* and *Different Verb* conditions

(b) Experiment 2: Priming under *Balanced* and *Skewed* conditions

Figure 4.6.5: Simulation results for Experiment 1 and 2 showing amount of priming versus increase in learning rate.

Let us first consider the Equal case, where the subject sees equal number of POs and DOs during training phase. In this case, varying λ does not affect structural priming under either Same Verb or Different Verb conditions (Figure 4.6.5(a)). Everything else being null (or default), the system shows 100% priming. This is only to be expected since any amount of learning (increased sensitivity) will affect both nodes equally under the equal-condition. That is, although we have structural priming in this case, it affects both nodes equally and does not bias the system to any construction.

Now let us consider the Unequal case. This case is more interesting. We notice that as learning rate, λ, increases, the amount of structural priming reduces quite significantly (the 'blue/triangle' and 'black/diamond' curves). Though it might seem unintuitive at first, this is exactly what is expected. The Unequal case in the experiment is set up so that the trials during testing phase use the opposite grammatical construction to the one used in the training phase. Hence as learning increases and leads to a bias towards the construction used in the training phase, it also leads to a bias against the construction used in the testing phase. This, in turn, leads to the observed reduction in priming. We can conclude that learning rate in the syntax layer, λ, affects only the 'Unequal case' during experiment 1 and is positively correlated to the amount of long-term structural repetition of the primed construction.

Lastly, we observe that λ does not have any effect on the amount of structural priming in the *Balanced / Skewed* experiment (figure 4.6.5(b)), and in the absence

of any lexical influence both cases show a high level (close to 100%)[14] of structural priming. Again, considering that both the Balanced and Skewed conditions are special cases of the Equal-case from the previous experiment, this is not a surprise. It does however confirm the belief that whatever difference we notice between the long-term priming of Balanced and Skewed conditions is down to lexical influence.

§ 4.6.3.2. **Rate of forgetting in binding nodes, δ_{bind}.—** Binding nodes link lexical nodes to syntactic nodes and therefore they form the medium through which lexemes can influence syntactic choice. They extend the idea of weighted connections between lemma nodes and combinatorial nodes in Pickering and Branigan (1998). The co-activation of lexical and syntactic nodes, during priming episodes, activates the connecting binding node. Once activated, this node shows hysteresis and remains in the active state for a particular length of time. If the target episode falls within this length of time (i.e. there are not too many filler items) then it acts as an *active link* between the lexical and syntactic node. An active link simply means that if the same lexical node is activated in the target episode (say, by an incomplete utterance: *John gave...*) the binding node boosts the input of the syntactic node activated in the priming trial. If, on the other hand, the link is not active, then the syntactic node receives no such boost. Through this mechanism lexemes influence syntactic choice.

But how long does this influence last? In our computational model this depends on how long the binding node remains *active*. Although the model resets the activation of each binding node at the beginning of a trial, it records the result of each episode in the internal memory of the node. Just like the variable K_{int} for lexical and syntax layers, the binding node has a variable E_{int} which undergoes learning, using the learning rule

$$\begin{aligned} E^t_{int} &= E^t_{ext} + \omega^t_i \\ \omega^t_i &= \delta_{bind}\omega^{t-1}_i + \lambda_{bind} \end{aligned} \tag{4.6.1}$$

The value of E_{int} acts as an internal input to the node. Just like K_{int} maintains a record of the input sensitivity of a WTA node, E_{int} maintains a record of the input sensitivity of a binding node. Thus E_{int} acts as a memory of all the previous episodes where learning took place. This memory means that activation of binding nodes depends not only on the immediately previous episode, but also on ones in the distant past. Just how far in the distant past depends on the rate at which E_{int} is learnt (λ_{bind}), and the rate at which it decays (δ_{bind}).

[14] This overall amount of priming can be decreased by varying other parameters, e.g. increasing the adaptation time, or decreasing the amount of hysteresis. That, however, does not impact the result that the difference between Balanced and Skewed conditions is independent of λ.

Therefore, we can re-frame our question about the duration of lexical influence to a set of questions about the activation of binding nodes:

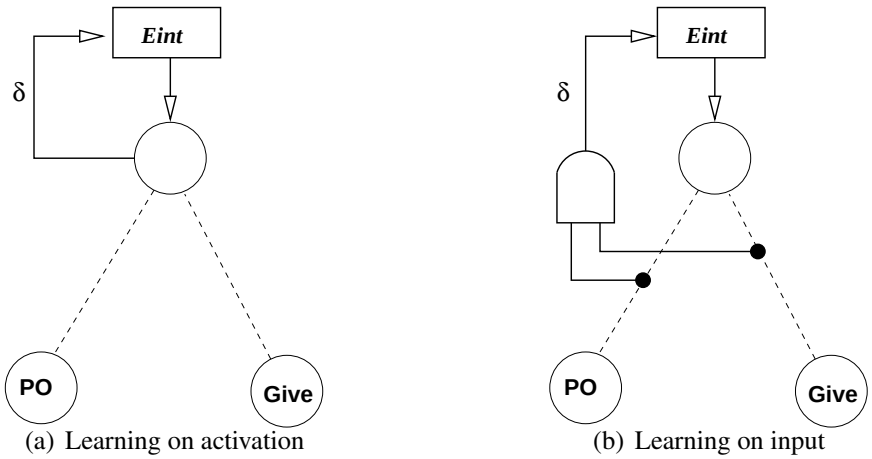
1. How does a binding node become active? That is, how does E_{int} learn? This relates the concept of choosing a *learning mechanism*.

2. How long does an active node remain active? That is, how does E_{int} decay? This is the concept of the *longevity* of activation.

3. How does the activation of binding nodes influence structural repetition?

The last question is the one we ultimately intend to answer: assuming we have long-term lexical-syntactic associations, how would they impact the structural repetition we observe on simulating the experiments of Kaschak and Borreggine (2008). But first we will have to specify the mechanism of learning and forgetting to see how those associations are created (activated) and destroyed.

Let us look at the learning mechanism first. There are two ways in which learning can take place in binding nodes. The first way is to learn (make a small change to E_{int}) whenever the binding node gets activated. This is similar to the mechanism that we have for learning in the lexical and syntactic nodes – the winner is rewarded by an increase in internal memory. The other way is to learn only when both the connected lexical and syntactic nodes are activated. That is, E_{int} learns the association between the lexical and syntactic nodes. These two different kinds of learning give two different longevities of activation. Let us look at each one in turn.



(a) Learning on activation (b) Learning on input

Figure 4.6.6: Learning on activation depends upon the activation of the binding node itself, while learning on input depends on the co-activation of both WTA nodes.

§ 4.6.3.3. **Learning on activation.—** With learning on activation, the model increases E_{int} each time the binding node gets activated at the end of priming episode.

$$\lambda_{bind} = \begin{cases} |\Lambda| & if \quad ON \\ 0 & if \quad OFF \end{cases} \tag{4.6.2}$$

Let us try to address each of the three questions raised above, under these conditions:

- ACTIVATION. A binding node can become activated if (a) both the input nodes are active at the end of priming episode, or (b) E_{int} is large enough to activate the binding node on its own.

- LONGEVITY. The longevity depends on both the rate of learning (λ_{bind}) and the rate of forgetting (δ_{bind}) (Equation 4.6.1). However, under the current mechanism of learning, once the node has been activated it will become self-sustaining. At the end of each episode, the sensitivity of an active node increases by a fixed amount ($|\Lambda|$) since its in an active state. Although this sensitivity would also show finite amount of forgetting (governed by δ_{bind}), this decay is not sufficient to turn the node OFF. It is not sufficient because the values of δ_{bind} and Λ are picked in such a way that the system shows cumulative learning. If δ_{bind} is too large, then the learning would completely decay at the end of each episode (figure 4.5.6(b)). Thus, if E_{int} is large enough, it will drive the binding node into an active state, even in the absence of external input. And since the node is activated, it will undergo learning, leading to a further increase in E_{int}. This means that once E_{int} has been driven above a threshold, then it will become self-sustaining and the binding node will always be active. In behavioural terms, this can be equated to developing a *permanent*[15] association between the lexical node and the syntactic node: every time the lexical node is activated, it would facilitate the associated syntactic node.

- INFLUENCE. Under this learning mechanism, lexemes would influence syntactic choice in one of three different ways: (i) if the decay rate (of E_{int}) is really fast then the longevity of binding nodes will be restricted to within an episode, and there will be no influence of previous episodes; (ii) If the decay rate is at an intermediate level then E_{int} might be driven above threshold; under input

[15]Of course, here we are considering this permanence in the context of a discourse. The human memory system is complex and there might be a host of other processes responsible for wiping the slate clean over longer periods of time.

scenarios where this can happen, we will get long-term influence of lexical nodes on syntactic choice; (iii) Finally, if decay rate is really slow, then E_{int} will quickly become self-sustaining and we will start getting lexical influence on syntactic choice for any association that has been activated even a few times during the training.

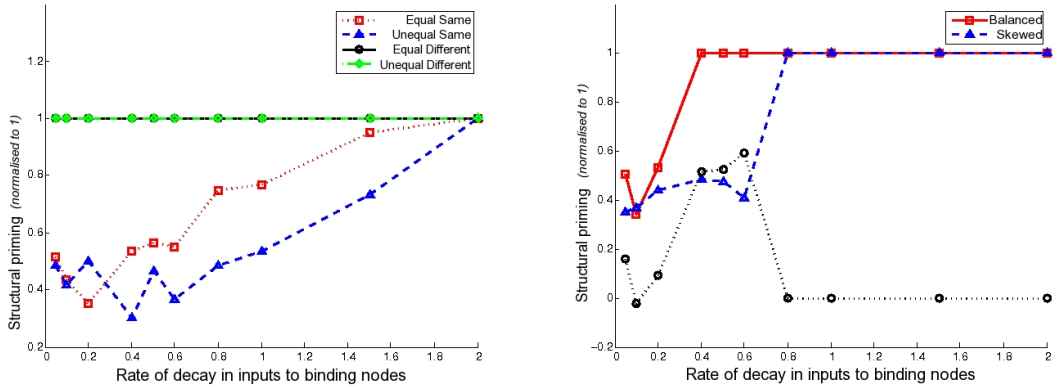
We are now in a position to see how variation in forgetting rate (for binding nodes) affects the results of the two experiments conducted by Kaschak and Borreggine (2008). We simulated the model for different rates of forgetting and made a counterintuitive finding: a system might show bias for a particular lexical association even if it is trained equally for two different associations. Specifically, the simulations find that in the experimental design used by Kaschak and Borreggine (2008) (figure 4.5.5), a verb might be equally associated with the two dative structures, and yet show a bias towards one of them. If this is the case, then the difference between equal and unequal cases (and balanced and skewed cases) no longer remains a measure of the amount of long-term priming. The premise used in the experimental design was that the Unequal case biases the verb towards a particular construction, while the Equal case has no long-term bias. By comparing the amounts of priming under the two cases, the experiment tried to reveal the long-term lexical enhancement of priming. When no difference was found, it was argued that lexical enhancement of priming is short-lived. But if the process of syntactic choice shows lexical bias even in Equal conditions, then one can no longer use the difference between the Equal and Unequal cases as a measure of long-term lexical enhancement of priming. Indeed, the results show that there might be long-term lexical enhancement of priming and yet no difference between the equal and unequal conditions. Let us have a closer look at these results.

The results for the two experiments are shown in figure 4.6.7. The figure plots the amount of structural priming versus an increase in the rate of forgetting. We choose an exponential rate of decay for E_{int} and we vary the order of the exponential (τ_δ):

$$\delta_{bind} = \frac{1}{\exp(\tau_\delta)} \tag{4.6.3}$$

The x axis shows the order of the exponential decay in the long-term learning rule for binding nodes – τ_δ – and the y axis shows the amount of structural priming for each of the four cases. As τ_δ increases, δ_{bind} decreases. And as δ_{bind} decreases, the input sensitivity, E_{int}, becomes more driven by current input. Finally, the greater the contribution of the current input, the faster the system forgets the contribution from

past episodes. So, in summary, values on the right (large τ_δ) represent very fast rates of forgetting while values on the left represent slow forgetting.[16]



(a) Variation of priming with rate of forgetting in binding nodes – *Same* and *Diff Verb* conditions

(b) Variation of priming with rate of forgetting in binding nodes – *Balanced* and *Skewed* conditions

Figure 4.6.7: Simulation results for Experiment 1 and 2 showing amount of priming versus increase in rate of forgetting.

Let us first look at the results for the Same Verb versus Different Verb experiment (Figure 4.6.7(a)). We observe that the rate of forgetting in binding nodes has no impact on the Different Verb conditions. This is not surprising. Under this condition, the verbs used during testing are disjoint from the verbs used during training. Consequently, the two phases have independent sets of binding nodes. And since binding nodes are the medium through which lexemes influence syntactic choice, there is no influence of lexical persistence on syntactic choice.

But what we are really interested in, is how the Same Verb conditions compare with the Different verb conditions and what happens on varying the rate of forgetting. This comparison can be made by looking at the Unequal-Same (red/square) and Equal-Same (blue/triangle) curves in Figure 4.6.7(a) and comparing them with the Different curves mentioned above. The first observation is that *both* the equal and unequal cases (under the Same Verb condition) show a decrease in the amount of structural repetition as a result of lexical influence. This decrease in repetition is evident from the values on the left in Figure 4.6.7(a). These values correspond to a slow rate of forgetting in the binding nodes – which, in turn, corresponds to longer lexical influence. Compared to the different case, structural repetition is down from 100% to around 60% for these

[16]It must be mentioned here that the value of persistence in structural memory itself was set to zero – so this is the impact on structural repetition caused independently by lexical influence.

values. These results show that when verbs are repeated and the links between lexemes and verbs forget slowly, then lexemes influence syntactic choice over a long term. As we mentioned above, the surprise lies in the fact that the amount of priming decreases in the *Equal* case. This was the case in which an equal number of PO and DO primes were provided in the training phase. So intuitively, there should be no bias towards any syntactic construction and hence no reduction in priming even if the lexical influence decays very slowly.

But one could argue that even if Equal cases show a reduction in priming, the amount of reduction might be less than the Unequal case. This would make it possible to distinguish the Equal and Unequal cases based on the difference in priming under the two cases. Therefore it is important to compare the quantity of this reduction in priming under the two cases. For a fast rate of decay (values on the right of figure 4.6.7(a)), both Equal and Unequal cases show very little and similar amount of decrease. For intermediate values, it seems that the Equal and Unequal cases, do indeed differ in the amount of reduction in priming, with Unequal cases showing a greater amount of reduction. However, the surprise lies at the left of the figure – cases where the decay is slow and forgetting is minimal. Under these circumstances, both Equal and Unequal cases show similar amount of decrease. Not only are the two cases similar for short-term lexical influence, but they are also similar for much longer lexical influence. Thus the similarity or difference in priming for equal and unequal cases gives us no indication of whether lexical influence is a short-term or a long-term phenomenon.

Let us also look at the results for the Balanced versus Skewed conditions (Figure 4.6.7(b)). These cases compare not only the overall bias towards alternative syntactic constructions, but also the specific bias of each verb towards each construction. Under balanced condition each verb is biased equally towards the two datives and under skewed conditions a verb is biased only towards one dative structure. We are interested in the difference in structural priming between the two conditions. Kaschak and Borreggine (2008) hypothesise that if lexical enhancement of priming accumulates over a series of episodes, then skewed conditions should carry the effect of the bias during training into testing, thereby reducing the amount of priming. On the other hand, the balanced condition should not develop any bias during training and therefore show no impact of long-term learning. Therefore, by comparing the two conditions, it should be possible to establish whether lexical enhancement of priming persists over a long period of time. Kaschak and Borreggine (2008) found no such difference. This led them to conclude that structural priming is independent of lexical influence in the long-term.

The simulation repeated the experiment for variable rates of forgetting in the binding nodes. The difference in priming for the two conditions is plotted as the black/circle curve in figure 4.6.7(b). Firstly it can be observed that this difference is close to zero for very rapid decay in binding nodes. This result is on the expected lines, and in agreement with Kaschak and Borreggine (2008): if lexical influence on structural choice decays very rapidly, then the balanced and skewed conditions will be similar. However, observe what happens as we decrease the rate of decay (i.e. move left). The difference follows a (inverted) *U-shaped* curve, reaching a maximum when E_{int} decays around $\exp -0.5$. If we keep decreasing the decay rate further, the difference between balanced and skewed conditions becomes close to zero – just like it was for a rapid rate of decay. Of course, the two scenarios are not identical. The difference is low for fast rate of forgetting because both conditions show close to 100% structural priming. And it is low for a slow rate of forgetting because both conditions have similar reduction in priming – down to 50%. But the essential result remains that Balanced and Skewed conditions show similar structural priming, not only when lexical influence decays rapidly, but also when it decays very slowly.

According to these results, if we observe that balanced and skewed cases show similar levels of structural repetition, then we could conclude, either (a) lexical influence is very short-lived, or (b) lexical influence is really long-lasting. Thus, simply observing that Balanced and Skewed cases show similar priming does not allow us to make any inference about the duration of lexical influence.

Just like the surprise in the Equal-Unequal experiment was really the fact that priming decreases under the Equal condition, the real surprise here is that priming decreases under the Balanced condition. Even though each verb is trained on stimuli that is unbiased towards any syntactic construction, it seems to develop a bias towards certain constructions. This bias leads to interference during the testing phase and decreases the amount of priming. It seems that, under this learning rule, presenting an unbiased stimulus, does not guarantee that the system will develop unbiased associations between lexical and syntactic constructs.

Let us try and understand why the system develops a bias towards one syntactic construct even when it receives both constructs equal number of times. Recall that the learning mechanism used to get this result is 'learning on activation'. Under this mechanism, binding nodes could have two different kinds of longevities: (a) dependent on rate of forgetting, if their internal memory, E_{int}, is below a threshold value, or (b) permanently active, if E_{int} is pushed above the threshold value. First let us look at

what happens when the rate of forgetting is very rapid. In this case, the binding nodes remain active only within an episode – i.e. between a priming trial and a target trial. This means there is no long term influence of lexical nodes on structural selection. Hence we get no reduction in priming (the values on the right of Figure 4.6.7(b)).

When we reduce the rate of forgetting to an intermediate value, some binding nodes might have their E_{int} value driven above the threshold. This is especially true in the Skewed case, where a verb is associated with only one kind of grammatical construction. The binding node representing this association will be repeatedly activated during the training phase. This will send the internal memory for that node, E_{int}, above the threshold and make the link between the verb and the grammatical construction permanent. On the other hand, the Balanced case associates a verb with both the constructions, thereby distributing the trials among the two different binding nodes. This means that the internal memory for binding nodes in this case is *not* pushed above the threshold and the link is not made permanent. As a result of this the Skewed case shows a greater reduction in priming for an intermediate rate of forgetting in binding nodes.

Finally, for very low rate of forgetting, even a couple of trials are enough to send E_{int} above threshold. This means that binding nodes activated during training phase in either Balanced or Skewed cases could have E_{int} above threshold. Crucially, in the Balanced training phase it is no longer sufficient that an association is activated the same number of times as the other. The exact sequence in which an association is activated becomes important. The sequence PO— PO—DO might send the PO link into active state, but that would not be the case if the subject receives PO—DO—PO. Thus the system might develop a bias towards one association even though both association were presented an equal number of times.

These biased association help us find an answer to the problem with which we began the section. At the start of the section, we noted that it is paradoxical that our model replicates thee results of Kaschak and Borreggine (2008). Their experiments were designed to detect the long-term influence of lexical information on syntactic decisions. We assumed in our model that the rate of decay is same in the syntactic layer and the binding nodes. This assumption meant that the lexical influence on syntactic decisions was long-lived. In spite of this long-term influence, our model replicated the results from Kaschak and Borreggine (2008), which they use to argue for the lack of such a long-term lexical influence. The solution to this paradox lies in an assumption implicit to the design of the experiments. This is the assumption that the *Equal* con-

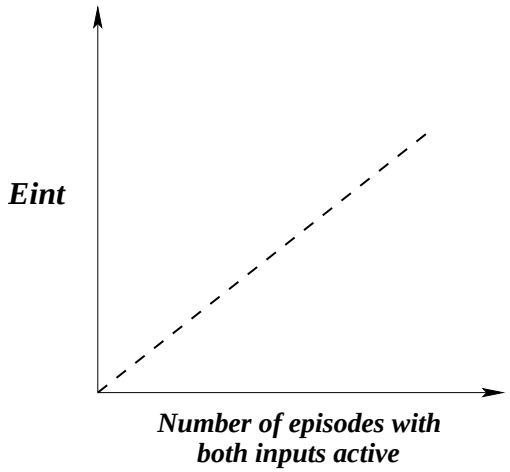


Figure 4.6.8: The value of E_{int} for a binding node is a linear function of the number of priming trials where input nodes are simultaneously activated.

dition, which presents the two dative structures an equal number of times during the training phase, leads to an unbiased association between the verb and the two dative structures.

Our analysis shows that even in the Equal condition the system might develop a bias towards one association. These biased associations for a slow rate of forgetting mean that comparing results under equal versus unequal stimuli and balanced versus skewed stimuli does not necessarily rule out a long-term lexical enhancement of priming. Therefore, a system could internally show long-term lexical enhancement, yet show no difference between Same Verb and Different Verb conditions, or Balanced and Skewed conditions. Of course, the underlying assumption of our model is the learning rule: fixed increment in the input sensitivity *whenever an association is activated*.

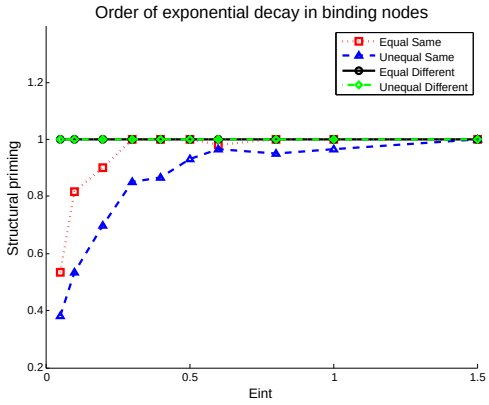
§ 4.6.3.4. **Learning on input.—** It is time to consider the alternative assumption: learning might not depend simply on the activation of a node at the end of an episode – it might depend on the co-activation of the two nodes connected to the binding node (Figure 4.6.6(b)). Therefore, this is a purely associative learning – lexical and syntactic nodes that are activated together more frequently have greater E_{int}. Thus the amount of increase in E_{int} is a linear function of co-activation. In comparison, E_{int} increased nonlinearly under the previous mechanism: learning could happen because of co-activation, or it could happen because E_{int} had passed a threshold. No such threshold value for E_{int} is applicable under this mechanism.

We also noticed that under the previous mechanism, the longevity of an activated

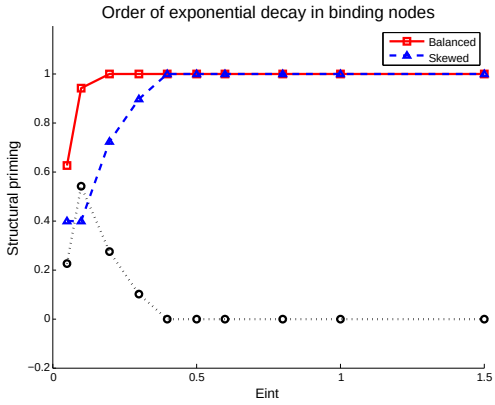
node could either depend completely on the rate of forgetting, or exhibit permanent activation – in which case it became independent of the rate of forgetting. In the current mechanism, on the other hand, since the learning is a linear function of co-activation, the longevity is directly dependent on rate of forgetting. And since we use an exponential function for forgetting, the dependence between the order of forgetting and longevity is exponential. That is, the order of decay in E_{int} is exponentially related to the duration for which a link remains activated.

Now that we know what to expect from this learning mechanism, let us see the results of the above experiments under these conditions (figure 4.6.9).

The results for the Same Verb versus Different Verb study (figure 4.6.9(a)) are similar to the results obtained by learning on activation. The Different Verb conditions are independent of variation in rate of forgetting while the Same Verb conditions show reduction in priming for slow decay. Just like figure 4.6.7(a), both Equal and Unequal cases show a reduction in priming – making effect size ($priming_{equal} - priming_{unequal}$) similar for slow and fast decay. The difference between the last mechanism and this lies in how structural priming increases with increase in rate of forgetting. While the rate of increase was roughly linear under learning on activation, here it seems to be asymptotic.



(a) Experiment 1: Priming under *Same Verb* and *Different Verb* conditions



(b) Experiment 2: Priming under *Balanced* and *Skewed* conditions

Figure 4.6.9: Simulation results for Experiment 1 and 2 showing amount of priming versus increase in rate of forgetting. This simulation uses the second learning mechanism: learning on input.

The results for Balanced versus Skewed conditions (figure 4.6.9(b)) differ significantly under this learning mechanism from the last. We notice that the U-shaped curve

has given way to a roughly exponential decay. This means that if this mechanism of learning is cognitively realised then the Balanced versus Skewed conditions will differ significantly if there is long-term lexical influence. This difference will decay exponentially as the duration of lexical influence becomes shorter. This is the result we expected from our discussion on linear mapping between co-activation and learning. Though it is expected, it is still an important result as it shows precisely how lexical representations influence syntactic choice. It also raises the need to conduct further psychological studies to examine which of these learning mechanism is more plausible.

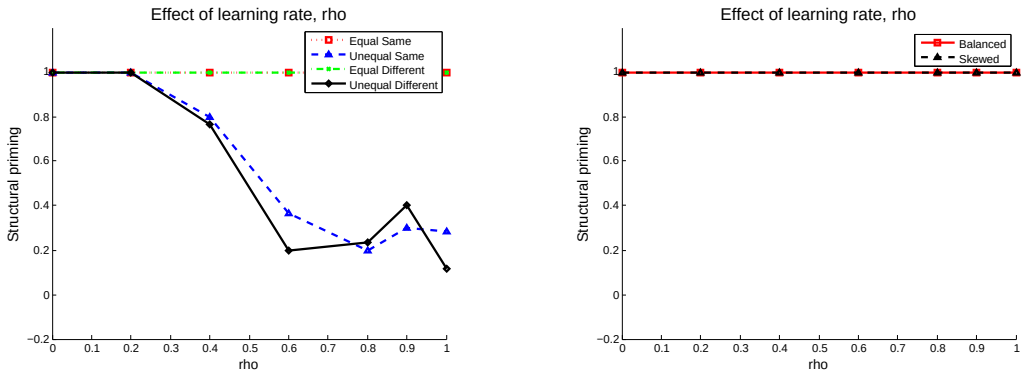
§ 4.6.3.5. **Time based forgetting.—** We saw above how different learning mechanisms can give different dynamics for lexical influence on syntactic choice. While we have varied the learning mechanism – i.e. conditions for learning – we have kept the learning rule the same. This learning rule is that of a *fixed* incremental adjustment to input connections at the end of each episode. In the case of binding nodes, we only make a fixed positive adjustment to the activated node(s). And in the case of WTA nodes we make positive adjustment to winning node, but negative adjustment to the losing nodes. This negative adjustment means that activation of a competing node, at the end of an episode, interferes with the long-term memory of the losing node. Therefore, the decay in memory is not strictly exponential and is influenced by the sequence of the input episodes. This is an interference-based account of decay in memory[17]

To check whether our results change if we change our learning rule to have strictly a time-based decay, we ran the two experiments using only positive adjustment to the winning node. That is, no (negative) adjustment is made to a node if it loses the target trial. The results for this are shown in Figure 4.6.10.

We observe that, in essence, the results remain similar to those obtained from interference-based learning rule (figure 4.6.5). Priming remains independent of variation in learning rate, λ, for the Equal cases. It decreases with increase in learning rate for Unequal cases. This again confirms that long-term learning in syntax layer is one of the factors responsible for lower priming in Unequal cases. Crucially, the reduction in priming for Same and Different verb conditions is similar. This means that long-term learning in syntax layer is not responsible for any difference between the two conditions.

All this is similar to the results from interference-based decay. What is different in

[17]Strictly speaking, this rule combines interference-based decay with time-decay since K_{int} decays both due to negative adjustments and due to an exponential decay at the end of each episode.



(a) Experiment 1: Priming under *Same Verb* and *Different Verb* conditions

(b) Experiment 2: Priming under *Balanced* and *Skewed* conditions

Figure 4.6.10: Simulation results for Experiment 1 and 2 showing amount of priming versus increase in learning rate. Simulation conducted with non-interference learning rule.

this case is that the decrease in priming, for Unequal cases, sets in later than it did for interference-based decay. At low rates of learning, the Equal and Unequal cases show similar amount of priming.

§ 4.6.3.6. **Prediction based learning.—** A characteristic feature of the learning rule, used so far, is making *fixed* adjustments after each episode. This choice was made because of our hypothesis: structural repetition in a discourse could be a result of trailing activation, and not a goal-directed learning (section 4.5.2). Now let us see what happens when we change our learning rule in another way: learning more when the input is surprising.

When a system shows variable amount of learning, depending on the input, it builds an internal model of its stimuli. The model makes a prediction and the system compares this prediction with external stimuli. If the prediction is incorrect, the system adjusts the model. This adjustment is proportional to the difference between the prediction and the external stimuli. The greater this difference, the more 'surprised' the model is and the larger the amount of learning it has to undergo. As discussed in Chapter 3, this is the learning mechanism that lies at the heart of the model presented in (Chang et al., 2006).

Two different kinds of experimental support is given for a goal-directed learning approach. The first one is the longevity of syntactic priming and the relatively short duration of lexical boost (Kaschak & Borreggine, 2008; Hartsuiker et al., 2008). The simulations in this chapter show that a trailing activation account (where the locus of

processes responsible for priming is internal) can replicate this experimental evidence. The other set of experiments used to support a goal-directed learning point to evidence from experiments studying priming in structures with variable frequencies (Hartsuiker, Kolk, & Huiskamp, 1999; Hartsuiker & Westenberg, 2000). These experiments find that low-frequency target structures benefit more from priming as compared to high-frequency target structures.

As we pointed in Section 2.3 (page 35), the claim for different amounts of priming for structures of different frequencies is controversial. However, for the sake of argument we decided to test our model with a different learning rule – one that is prediction-based and learns more for surprising information. While we adjust the learning rule, the architecture of the model remains the same – lexical and syntactic information is represented independently in WTA layers and connected through a set of binding nodes. Thus changing the learning rule allows us to challenge the assumption that the results of our simulation rely on a fixed learning mechanism.

Now, let us see how we can implement a prediction-based learning in our model. The computational model developed by Chang et al. (2006) makes predictions through the inbuilt sequential recurrent network. Our model does not have such an inbuilt mechanism. If we want it to perform predictive learning, our model would have to use an approximation to perform these predictions. A good heuristic for this is learning according to the surprise value of input information. There is previous evidence (Hale, 2006) that the surprise value of input information is a good estimation of the cognitive load during language comprehension. It relies on the intuition that our cognitive system is better at registering surprising information than it is at registering banal information.

Our next task was to identify a variable in the system that estimates the surprise value of a structural construct. For this, we used the frequency with which the construct appears in the training phase. We change the learning rule to make large adjustments when it comes across a construct of low frequency, and small adjustments when it comes across high-frequency structures. The new learning rule becomes,

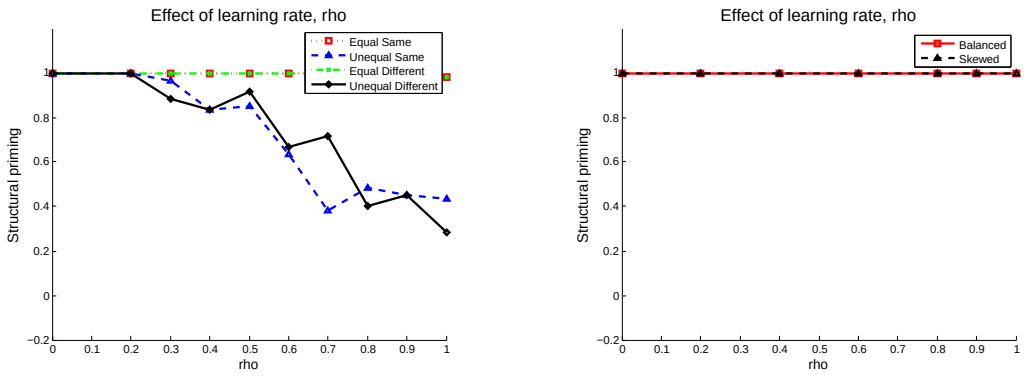
$$\begin{aligned} \omega_i^t &= \delta\, \omega_i^{t-1} + \lambda \\ \lambda &= \begin{cases} \frac{+\Lambda}{1+|\omega_i^t|} & if \quad ON \\ \frac{-\Lambda}{1+|\omega_i^t|} & if \quad OFF \end{cases} \end{aligned} \tag{4.6.4}$$

One can compare this learning rule with the previous learning rule (Equation 4.5.4). The crucial difference is that the amount of learning, λ, depends on the current sensitivity of the node, ω_i^t. If a node has a high sensitivity, then the amount of learning is

low. On the other hand, a node with low sensitivity (i.e. large surprise value) will have larger adjustment, λ.

The assumption here is that more frequent constructs are going to be more likely during the testing phase. If a construct appears more frequently during the training phase, the sensitivity of the construct will be large and therefore it will show a small amount of adjustment during the testing phase. Consistent with this approach, the frequency-based results mentioned above (Hartsuiker et al., 1999; Hartsuiker & Westenberg, 2000) do indeed compare the relative frequencies of targets appearing during a pre-experimental baseline with frequency of targets appearing during the experimental phase. Also, this heuristic for measuring surprise is not too different from the error-based learning in the model proposed by Chang et al. (2006) which also internalises the frequency structure of the environment during training.

We plugged in this learning rule into our system and simulated it for the experimental design for Model ②. Again we vary the learning rate, λ, and simulate the computational model for the two experiments from Kaschak and Borreggine (2008). The results are shown in Figure 4.6.11.



(a) Experiment 1: Priming under *Same Verb* and *Different Verb* conditions

(b) Experiment 2: Priming under *Balanced* and *Skewed* conditions

Figure 4.6.11: Simulation results for Experiment 1 and 2 showing amount of priming versus increase in learning rate. Simulation conducted with prediction-based learning rule.

Let us look at the results for the Balanced versus Skewed conditions first (Figure 4.6.11(b)). These results are exactly same as those for the fixed learning rule: neither Balanced nor Skewed condition depends on increase in rate of learning in syntax layer. The results for Same and Different verb conditions are more interesting. Just like for the fixed learning rule, the Equal cases for both conditions show no depen-

dence on learning rate. The Unequal cases however seem to decrease in a rather non-monotonic manner with increase in learning rate. Also the difference begins to emerge later (around $\lambda = 0.5$) than under fixed learning rule (figure 4.6.5(a)).

But besides these minor differences, the results for Same versus Different conditions are similar under prediction based learning. The Unequal cases show lesser priming than Equal cases for large learning rate; both Unequal-Same and Unequal-Different show similar amounts of reduction in priming; and the general trend is for a decrease in priming as learning rate increases. These results suggest that the architecture for the current model can just as easily be used for prediction-based learning, at least, as far as the experiments of Kaschak and Borreggine (2008) are concerned. The results do not arbitrate between a goal-based learning account and a fixed amount of learning, but they do suggest that either could lead to the temporal properties of syntactic priming and lexical enhancement of this priming.

4.7 Conclusion

We started this chapter with the knowledge that people repeat syntactic structure while speaking. We also knew that this repetition can be enhanced by repeating the verb. Finally, we knew that syntactic repetition and the lexical enhancement of this repetition show different temporal properties. What we did not know was whether the trailing activation model can account for these different temporal properties.

In this chapter we have developed a computational model that shows that a trailing activation account can indeed show these variable rates of decay in syntactic repetition and its lexical enhancement. In the process, we have embellished the trailing-activation account itself. At a computational level, we have seen how we can formalise the notion of trailing activation through the mechanism of hysteresis in a dynamical system. We have also seen that such a system needs to perform incremental learning if it has to accumulate information over a sequence of episodes. Researchers frequently make the distinction between trailing activation and implicit learning. In this chapter, we have bridged this gap by developing a system that that shows incremental (implicit) learning and yet is compatible with the principles of trailing activation. Instead, we have redrawn the boundaries for learning mechanisms between a goal-directed, supervised, variable amount of learning on the one hand and a cognitively inspired, unsupervised, fixed amount of learning on the other. We have seen that experimental findings that were believed to be compatible with only the former form of learning are also compat-

ible with the latter.

What consequences does this have for the study of language production? To begin with, the fact that priming and its temporal properties can be explained through a trailing activation account shows that repetition might not be a consequence solely of the processes of language acquisition, a claim that has been made by Chang et al. (2006). Let us develop this argument a bit further.

Interlocutors might have two reasons for learning during a dialogue. The first reason is to develop their knowledge of how the language works. This account assumes that the language expertise lies in the linguistic community that the interlocutors interacts with. Interlocutors are embedded in this linguistic community and try to build an internal model of language. Each time they hear a phrase, interlocutors try to adjust their internal model to make it compatible with the phrase. This adjustment is one reason why they repeat syntactic structure of recently heard utterances.

The other reason for learning is to develop a discourse model. The linguistic signal is sequential and is spread through time – across phonemes, words, phrases, sentences and even conversations. Interlocutors want to integrate this information that is spread in time and the only solution is to rely on memory – i.e. to learn. Each phoneme, word, phrase and such leaves a memory trace when it is processed by the linguistic system. Each time interlocutors want to produce a sentence, they do not construct it from the scratch, but rely on previous memory traces. Not only does this integration of memory traces save system resources (through reuse), but it is made necessary because linguistic signals are spread over time. The last part of the puzzle is that this integration comes with interference. When interlocutors face linguistic choices (such as POs or DOs; Actives or Passives), the memory traces bias these choices in certain directions. Reusing a DO that an interlocutor has recently heard saves resources, but it also ensures integration with the speaker's discourse model. Reusing a memory trace ensures that the speaker and listener are aligned (Pickering & Garrod, 2004).

But memory is a complex system. It serves not just language but other aspects of cognition too – vision, motor skills, emotions and others. It is also implemented through a biological system – populations of neurons and their connections. As a consequence, the nature of cognitive implementation affects memory which, in turn, affects repetition in language. In this chapter, we saw how repetition might decrease with time as a consequence of fatigue or adaptation in populations of neurons. We also saw that while structural representations might be implemented through one type of a dynamical system (WTA), the lexical influence on structural decisions might be medi-

ated through a different kind of a system (STM). The two different types of systems show different kinds of memories (or fatigue) and therefore might explain the experimental observation that structural repetition persists longer than lexical enhancement of this repetition.

Memory is complex in another way. When sequence of episodes of language processing accumulate over a period of time, certain sequences might be more important than others. We noticed this in our analysis of incremental learning. Our system developed a bias towards certain choices even though it was presented both those choices an equal number of times. The reason is that the learning mechanism that integrates the information is nonlinear: PO—PO—DO—DO $\neq$ PO—DO—PO—DO. The sequence in which information is presented might be important. Kaschak and Borreggine (2008) present subjects with verbs that are associated with PO and DO structures equal number of times and expect that subjects form equal associations with both structures. The simulations performed with 'learning on activation' rule challenge this assumption. Even if every episode has equal amount of attention drawn to it, some sequence of episodes leave a larger trace on memory than others. If integration in memory was linear then all episodes will be equal, but if it is nonlinear some episodes might be more equal than others.

Lastly, this chapter explores a novel method for studying priming – dynamical systems. It shows that linguistic choices can be modelled as dynamics of a system near bifurcation. A formal system allows us to investigate the assumptions underlying different hypotheses. It exposes a limited set of parameters that allow us to tune the system so that it gives results consistent with behavioural data. The values of these parameters then provide insight into the role of different cognitive processes responsible for priming.

5 A novel framework for comprehension and production

5.1 Motivation

Until now we have considered theoretical models that replicate experimental findings about syntactic priming and lexical boost. However, we have not considered the plausibility of these models. Our goal was to search for computational mechanisms that can explain how syntactic priming changes with time and with lexical overlap. Having established these computational mechanisms, we now turn to developing a system that is more complete. Such a system not only needs to provide an account of structural choice, but also needs to specify when this choice is made during comprehension and production and how these processes can be plausibly represented.

The model that we developed in the last chapter has several limitations that make it cognitively implausible, representationally underspecified or procedurally incomplete:

- LOCALIZED REPRESENTATIONS. The model in the last chapter represents lexical and syntactic constructs as nodes, rather than patterns of activation over a network. Such a localized representation has several limitations such as catastrophic degradation with noise, hard capacity limit and a lack of generalisation (Plate, 1997)[1]. Importantly, localized distributions do not allow coarse coding and therefore do not allow us to study interference between different linguistic constructs. For example, we cannot use the existing model to study how syntactic priming changes with conceptual overlap. Thus, we need to replace the localized representation of the existing model with a distributed representation

[1] However, see Thorpe (1995) for some rebuttals to critiques of localized representations.

at each layer.

- LEARNING ALGORITHM. Two different kinds of learning mechanisms operate in the existing model: (i) hysteresis in WTA and STM networks which is responsible for priming when the target immediately follows the prime, and (ii) incremental adjustment to input sensitivity of nodes, which is responsible for cumulative effect of priming over several prime trials. In this chapter, we would like to explore whether we can find a single learning mechanism that can seamlessly explain both immediate and cumulative effects of priming. In the interest of plausibility, we would also like this learning mechanism to be previously studied for physiological memory networks.

- SEMANTIC INFLUENCE. The model in the previous chapter does not include any conceptual representations. It only studies structural choice under lexical influence. This is a shortcoming of the model since semantic representations drive syntactic choice during production and are extracted using syntactic relations during comprehension (Levelt, 1989; Levelt et al., 1999). In this chapter, we would like to correct this limitation and extend the system to include conceptual representations and explore how these representations might influence syntactic choice.

- EXPLICIT PROCESSES OF COMPREHENSION AND PRODUCTION. In the last chapter, we simulated comprehension by providing a large external stimulus to the nodes and production by allowing the nodes to settle into a stable state on their own. While this implementation allowed us to concentrate on the mechanism of syntactic choice, it leaves open questions as to how the syntactic and lexical nodes receive this large input during comprehension and what is the sequence in which lexical, syntactic and semantic decisions are made by the system. In this chapter we intend to make the system more complete – i.e. we would like to specify the sequence of processes and the flow of activation during comprehension and production. We would especially like to see how the two might overlap and whether one process can prime the other. These are important questions in the study of dialogue where this overlap between comprehension and production processes could be a mechanism of achieving alignment between interlocutors (Pickering & Garrod, 2004).

Before we proceed, it must be mentioned that when we talk about a *complete* sys-

tem we do not mean a system that can understand and generate natural language from scratch. Such an enterprise would not only be overly ambitious, but also largely irrelevant to the processes of syntactic priming. Instead, our goal is a more modest one of extending the model presented in the last chapter so that we can investigate how syntactic representations might be generated from a sequence of words during comprehension and to see how concepts are arranged into one sequence or the other during production. We will limit ourselves to studying the dative structure and would like to specify how the system understands or produces such a structure and what role memory plays in each of these processes.

5.2 Theoretical extensions

Addressing the first three limitations, noted above, requires three extensions to our theoretical repertoire, and the last limitation requires a change in architecture of our model. In this section we present theoretical concepts required to extend our model and in the next, we present a new architecture.

§ 5.2.1 Binding for distributed representations

A chief obstacle in moving from localized to distributed representations is binding these distributed representations together. Localized coding has a one-to-one mapping between a concept and its representation. Each concept is represented by a node and each node represents one concept. This makes the task of binding concepts together quite simple. In the previous chapter, for example, we saw that binding nodes record the association between lexical and syntactic nodes. Models ①, ② and ③ implement static binding which means that the association between a lexical and a syntactic construct is explicitly encoded by a node. In Model ③ (figure 4.5.1 on page 138), for example, each lexical node is connected to both the combinatorial nodes through two binding nodes. Since there are four lexical nodes and two combinatorial nodes, we have a total of eight binding nodes, each binding node representing a lexical-syntactic association uniquely specified by the attached lexical and syntactic nodes.

A distributed code, on the other hand, represents a concept as a pattern of activation over a group of nodes. Each concept can be represented by several nodes and each node can play a part in representing several concepts. Because there is no longer any one-to-one correspondence between a concept and a node, binding two concepts now becomes

a non-trivial problem. It is no longer clear which nodes to include in a binding, whether overlapping representations will lead to noise during unbinding, etc.

Fortunately, this problem is not unique to the representation required for our model and has been well studied in another context – that of representing relational structure in connectionist networks (Plate, 1997). In order to represent a sentence such as *Mary spied on John yesterday*, a connectionist network needs to represent not only each of the concepts *Mary*, *spied on*, *John* and *yesterday*, but also bind them together as part of one sentence. Furthermore, the connectionist network needs to perform this binding in such a way that this sentence is not confused with *John spied on Mary yesterday*. Thus, the connectionist network needs to represent not just the constituent concepts, but also their relational structure. If each of the concepts is represented as a distributed pattern of activation, then the network needs to perform binding over these distributed representations and disambiguate this binding from other possible bindings over these concepts.

Three major schemes have been proposed to perform binding for distributed representations. Pollack (1990) has developed a Recursive Auto Associative Memory (RAAM) for representing the compositional structure of sentences. Plate (1995, 1997) proposed *Holographic Reduced Representations* (HRRs) which bind two representations by performing circular convolution (defined below) over these representations. Lastly, Smolensky (1990) proposed that binding between patterns over two vectors can be calculated by finding the tensor product between these vectors.

When we consider the network architecture in the next section, we will see that there are two kinds of bindings that we would like to represent – one within a representational layer and another between two layers. Both HRRs and tensor product binding schemes are suitable for each of these bindings. We briefly present both these schemes below and discuss reasons for choosing the tensor product as the preferred method of encoding bindings in our system. The third scheme – RAAM – is a more complex scheme used to learn reduced representation for tree structures (Pollack, 1990). Our interest, for the current purposes, is not to represent such a tree structure, but a simple binding between roles and fillers. Therefore, in the interest of simplicity, we do not pursue this binding mechanism any further.

§ 5.2.1.1. **Holographic Reduced Representations.—** Holographic Reduced Representations were developed by Plate (1995) to represent the compositional, tree-like structure of language using distributed representations over a connectionist network.

Plate (1995) discusses how it is difficult for matrix memories, such as auto-associative memories (e.g. Hopfield networks) and hetro-associative memories (e.g. feedforward networks), to represent the recursive associations between constituents of sentences. As a solution to this problem, they propose a representation scheme based on *holographic* memory (Willshaw, 1981), instead of matrix memory.

The idea behind holographic reduced representations stems from the mathematical operation of *circular convolution*. The convolution, or more specifically aperiodic convolution, of two vectors **x** and **h** is defined as:

$$\begin{aligned} \mathbf{y} &= \mathbf{x} * \mathbf{h} \\ y(i) &= \sum_{k=-\infty}^{\infty} x(k)\, h(i-k) \end{aligned}$$

We can store the association between two vectors **x** and **h** using their convolution **y**, which serves as the memory of the association. At a later point, we can retrieve the vector associated with the vector **x**, given the convolution memory **y**, by performing *deconvolution*, which is an algorithm that can reverse the effects of convolution. Thus, convolution can be used to store the binding between patterns of activation over two vectors.

However using convolution to perform binding for compositional structure has two problems. Firstly, deconvolution of two vectors is not a straightforward process and leads to noise in the retrieved memory. Secondly, if each of **x** and **h** are of length N, then their convolution **y** is of length $2N-1$. This means that if we want to use the process of convolution recursively, such that we want to bind the output of a convolution to another vector, then the dimensionality of the vector keeps expanding.

Plate (1995) propose that this problem of expanding dimensionality can be avoided if we use circular convolution as the binding operation, instead of aperiodic convolution. The circular convolution between **x** and **h**, each of length N, is defined as:

$$\begin{aligned} \mathbf{y} &= \mathbf{x} \circledast \mathbf{h} \\ y(i) &= \sum_{k=0}^{N-1} x_N(k)\, h_N(i-k) \end{aligned}$$

where the subscript N on x and h denotes that the indexes are modulo-N – i.e. they wrap around when the index is larger than N. Crucially, the circular convolution of two vectors of length N is another vector of the same length N, thereby avoiding the problem of expanding dimensionality.

The second crucial insight provided by Plate (1995) is that the circular cross-correlation operation, $\star$, is an approximate inverse of the circular convolution operation, which means that instead of relying on the difficult algorithm of deconvolution, this scheme can retrieve memories using the simple operation of circular cross-correlation:

$$\begin{aligned} \mathbf{y} &= \mathbf{x} \circledast \mathbf{h} \\ \mathbf{y} \star \mathbf{x} &\approx \mathbf{h} \end{aligned}$$

where the circular cross-correlation of two vectors $\mathbf{y}$ and $\mathbf{x}$ is an operation that is quite similiar to the circular convolution. Just like convolution, the correlation operation takes two input signals and finds an output signal. Howerver, unlike the convolution operation, the correlation operation tries to detect a known signal $\mathbf{x}$ in a noisy input $\mathbf{y}$. Formally, the circular cross-correlation between two vectors $\mathbf{y}$ and $\mathbf{x}$, each of length N, is defined as:

$$\begin{aligned} \mathbf{h} &= y \star x \\ h(i) &= \sum_{k=0}^{N-1} y_N(k)\, x_N(k+i) \end{aligned}$$

where the subscript N on y and x again denotes that the indexes are modulo-N. Because this convolution-correlation scheme works on vectors of the same length (N), a number of vectors can be hierarchically combined together to give a single vector. This vector is the compressed, or *reduced* representation of the binding between each of these vectors and hence the name Holographic Reduced Representations.

The last piece of the puzzle comes from the illustration that different convolutions can be superimposed upon one another through summation. Superposition allows several vectors to be stored in the same memory. While superposition inevitably leads to interference, Plate (1995) argued that HRRs can be complemented with an auto-associative memory (such as a Hopfield network), which can perform the "clean-up" of retrieved memories and improve recall.

An example (taken from Plate (1997)) illustrates the use of HRRs for storing and retrieval of compositional structure. Consider the proposition *Spot bit Jane, which caused Jane to run away from Spot.* This proposition (say $\mathbf{P}_{cause}$) can be divided into two constituent propositions: *Spot bit Jane* (say $\mathbf{P}_{bite}$) and *Jane ran away from Spot* (say $\mathbf{P}_{flee}$). Each of the propositions can be represented as a circular convolution of

different concepts and their *roles* in the proposition:

$$\begin{aligned}\mathbf{P}_{bite} &= <\mathbf{bite} + \mathbf{bite}_{agt} \circledast \mathbf{spot} + \mathbf{bite}_{obj} \circledast \mathbf{jane}> \\ \mathbf{P}_{flee} &= <\mathbf{flee} + \mathbf{flee}_{agt} \circledast \mathbf{jane} + \mathbf{flee}_{from} \circledast \mathbf{spot}>\end{aligned}$$

where we use circular convolution to bind the concepts **spot** and **jane** to the roles of $\mathbf{bite}_{agt}$, $\mathbf{bite}_{obj}$, $\mathbf{flee}_{agt}$ and $\mathbf{flee}_{from}$. In addition, each binding in a proposition is superimposed with other bindings in that proposition through summation. Finally, the two propositions can be (recursively) bound to their respective roles in the higher level proposition, $\mathbf{P}_{cause}$:

$$\begin{aligned}\mathbf{P}_{cause} = <&\mathbf{cause} + \mathbf{P}_{bite} + \mathbf{P}_{cause} \\ &+ \mathbf{cause}_{antc} \circledast \mathbf{P}_{bite} + \mathbf{cause}_{cnsq} \circledast \mathbf{P}_{flee}>\end{aligned}$$

Thus the single vector $\mathbf{P}_{cause}$ functions the memory for the entire proposition, "Spot bit Jane, which caused Jane to run away from Spot," including all its constituent concepts and their roles in the proposition.

Plate (1995) also showed that such a vector can be used at a later point of time, to obtain any of the sub-parts, through the process of circular cross-correlation. If, for example, one wishes to find who did the biting, one can do so by evaluating $(\mathbf{P}_{cause} \star \mathbf{cause}_{antc}) \star \mathbf{bite}_{agt}$, i.e. one first finds the antecedent of $\mathbf{P}_{cause}$, which should give $\mathbf{P}_{bite}$ and then the agent in $\mathbf{P}_{bite}$, which should give **spot**.

This example illustrates that HRRs provide a binding mechanism for distributed representations that can be used for storing the compositional structure of linguistic expressions. Given two vectors, the binding can be calculated immediately, as this mechanism does not entail any learning. Most importantly, this mechanism provides a straightforward unbinding mechanism through circular cross-correlation, which can be used to perform retrieval of one constituent, given the other constituent and the binding.

§ 5.2.1.2. **Tensor product binding.—** We have seen that one mechanism of binding distributed representations is the mathematical operation of circular convolution. An alternative binding mechanism is provided by an alternative mathematical operation – the tensor product. The tensor product between an n-dimensional vector $\mathbf{a}$ and an m-dimensional vector $\mathbf{b}$, is the nm dimensional vector $\mathbf{a} \otimes \mathbf{b}$, whose elements are all possible products $a_i b_j$ of an element of $\mathbf{a}$ and an element of $\mathbf{b}$.

Smolensky (1990) showed that tensor products can be used to represent symbolic structure in connectionist networks. For doing this, a symbolic structure should be

expressed as a conjunction of *roles* and *fillers* – i.e. each symbolic structure is factored out, or *decomposed*, into a set of roles and fillers. Smolensky (1990) proposed that such role-filler decompositions can be used to represent different kinds of symbolic structures. For example, strings can be seen as an array of characters. Each element of the array contains a role and a filler. In the case of a string, the role is just the position of the character in the string and the filler is the character itself. Similarly, a proposition can be seen as a set of predicates, each of which is a conjunction of roles and fillers (as we saw in the "Spot bit Jane" example above).

After decomposing a symbolic structure into a conjunction of fillers and roles, the next step is to find a connectionist representation of such a conjunction. In the scheme presented by Smolensky (1990), a connectionist network represents a conjunction through superposition, i.e.

$$\psi\left(\bigwedge_i p_i\right) = \sum_i \psi(p_i)$$

where $\wedge$ is the symbol for conjunction, p_i is the i^{th} proposition and ψ is a function that maps a symbol (such as p_i) onto its connectionist representation. So the above rule implies that the connectionist representation of a conjunction of propositions is the sum of the connectionist representations for each of the individual propositions. Thus a symbolic structure, which is a conjunction of a set of roles and fillers, can be represented in a connectionist network as a sum of the connectionist representations of each role and its filler.

The last step is to find a connectionist representation of each of the roles and fillers. This is where the operation of tensor product comes in. Each role and its filler can be represented as a pattern of activity over a set of nodes. This pattern of activity over a set of nodes can then be encoded as a vector – which is nothing but a tensor of rank one (as we will discuss below). If $\mathbf{f}$ is a vector representing the filler f (i.e. $\mathbf{f} = \psi(f)$) and $\mathbf{r}$ is the vector representing the role r (i.e. $\mathbf{r} = \psi(r)$), then the role-filler association (or binding) can be represented by the tensor product of the two vectors, $\mathbf{f}/\mathbf{r} = \mathbf{f} \otimes \mathbf{r}$, where $\mathbf{f}/\mathbf{r}$ represents the association between the vectors $\mathbf{f}$ and $\mathbf{r}$.

Putting it all together, we can say that a symbolic structure which can be decomposed into a conjunction of role-filler bindings can be represented in a connectionist framework through a sum of tensor products between roles and their corresponding

fillers, which is elegantly conveyed by Smolensky (1990) in the equation:

$$\psi\left(\bigwedge_i f_i/r_i\right) = \sum_i \mathbf{f}_i \otimes \mathbf{r}_i$$

The left side of the equation states the problem – finding a connectionist representation for a conjunction of role-filler combinations. The right side of the equation delivers the solution – the sum of tensor products of roles and their corresponding filler vectors.

This scheme of representing symbolic structure in connectionist framework can be more clearly illustrated through a diagram. Figure 5.2.1 shows a vector $\mathbf{bite}_{agt}$, which encodes a particular role, along the x-axis and a vector **spot**, which is the filler for the role, along the y-axis. The tensor product is simply the outer product of the two vectors, calculated by multiplying each of the elements of the two vectors together. This tensor product gives a matrix, which is shown in the centre of the figure.

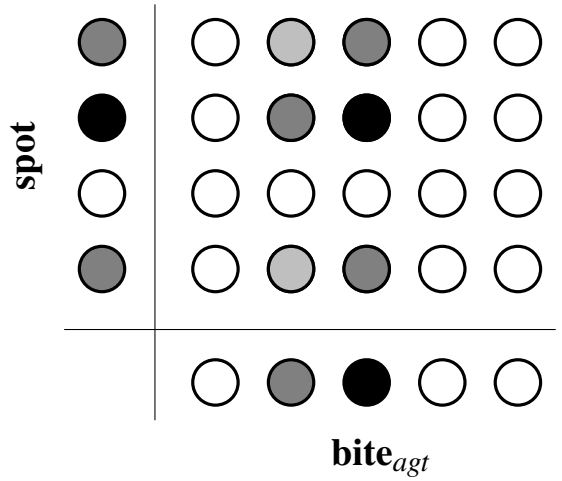


Figure 5.2.1: [Tensor product] The tensor product of $\mathbf{bite}_{agt}$ and **spot** is calculated by finding the outer product of each of the elements of the two vectors. Each vector is shown along an axis in the 2-dimensional space and their tensor product is shown in the middle. The activity of each node is characterised by its brightness, with darker nodes showing high activity and brighter nodes show a low value of activation. (Adapted from Smolensky (1990))

While the above analysis demonstrates that it is possible to represent symbolic structure in a connectionist framework, it does not tell us whether this coding scheme is going to be efficient. Once we store a set of role-filler associations in a connectionist network through tensor product binding, would we be able to retrieve the filler corresponding to a role at a later point of time? How do different vectors stored in such

a memory interfere with each other and can we use this memory to store bindings of more than two vectors? In order to answer these questions, Smolensky (1990) presented a detailed account of the properties of tensor product bindings that make this scheme suitable for storing symbolic structures. We present a brief outline of a couple of properties relevant to our implementation; the interested reader is directed towards Smolensky (1990) and Smolensky (1987) for a more detailed discussion.

- UNBINDING. Unbinding is the procedure of retrieving one element of an association, given the other element and the binding (or memory) that associates the two elements. We saw above that the circular cross-correlation operation, $\star$, provides an approximate unbinding mechanism for HRRs. Similarly, if we want to use the tensor product as a memory, then we need to establish an unbinding mechanism to perform retrieval from the memory. Smolensky (1990) proved that, provided the vectors representing roles in a structure are linearly independent[2], each of the fillers f_i bound to a role r_i can be unbound from the tensor product representation, with complete accuracy, using an unbinding vector u_i. While this unbinding operation is theoretically possible, to do so one must first find the *unbinding vector* corresponding to each role. The inner product of this unbinding vector with the tensor product binding gives the required filler:

$$\left(\sum_j \mathbf{f}_j \otimes \mathbf{r}_j\right) \cdot \mathbf{u}_i \;=\; \mathbf{f}_i$$

  More importantly, Smolensky (1990) also showed that when such an unbinding vector u_i is not available, one can use the role vector r_i itself to perform the unbinding. For obvious reasons, this unbinding procedure is called the *self addressing* unbinding procedure. The procedure of calculating the unbinding is similar to the one presented for the vector u_i above: one needs to find the inner product of the role vector with the tensor product representation. In this case the roles do not even need to be linearly independent. However, there is a downside: noise due to interference. Smolensky (1990) showed that the closer two role vectors are, the more likely there will be interference from one to the other. But if the vectors are orthogonal, then this noise disappears. Specifically, unbinding

[2] A set of vectors $\mathbf{r_1}, \mathbf{r_2}, \ldots, \mathbf{r_n}$ are linearly independent if they cannot be expressed in the form $a_1\mathbf{r_1} + a_2\mathbf{r_2} + \ldots + a_n\mathbf{r_n} = 0$, where $a_1, a_2, \ldots, a_n$ are non-zero constants.

role $\mathbf{r}_i$ from a sum of role-filler bindings $\mathbf{f}_j \otimes \mathbf{r}_j$, gives the result:

$$\left(\sum_j \mathbf{f}_j \otimes \mathbf{r}_j\right) \cdot \mathbf{r}_i = \mathbf{f}_i + \cos\theta_{ji} \frac{\|\mathbf{r}_j\|}{\|\mathbf{r}_i\|} \tag{5.2.1}$$

where θ_{ij} is the angle between vectors $\mathbf{r}_i$ and $\mathbf{r}_j$. If the vectors $\mathbf{r}_i$ and $\mathbf{r}_j$ are orthogonal then the second term becomes zero, leading to zero noise.

- TENSORS AND VECTORS. Tensors are generalisations of vectors and tensor products are generalisations of the outer product of vectors. The generalisation comes when one moves from one-dimensional indexing to multi-dimensional indexing. The word *tensor* signifies an array of numbers that can be indexed. The number of indices required to pick an element of a tensor is called its *rank*. A vector is a tensor of rank one (i.e. one index is required to pick an element in a vector) and a tensor product of two vectors gives a matrix, which is a tensor of rank two. We can continue along this path and find a tensor product of three vectors $\mathbf{a} \otimes \mathbf{b} \otimes \mathbf{c}$, which will give us a tensor of rank three. This generalisation is useful as it removes the restriction of being able to bind only two vectors at a time. For example, if we have to bind lexical, syntactic and phonological representations together we can code each of the representations into a vector and find a tensor product of the three vectors.

§ 5.2.1.3. **Choosing a binding mechanism.—** Both tensor products and HRRs are capable of binding distributed representations of symbolic structure. HRRs are particularly attractive because of their scalability: the binding of two vectors of length n can be stored in another vector of the same length n. If we have to perform the binding operations repetitively over symbolic structure, for example when the symbolic structure is hierarchical, the scalability of HRRs means that we do not face the problem of expanding dimensionality. On the other hand, tensor product binding retains complete information about the components of the binding, allowing us to retrieve each component from the binding with complete accuracy. Tensor products can represent hierarchical structures, but the size of the patterns increases exponentially with the depth of the hierarchy (Plate, 1997). In fact, HRRs and tensor-product binding differ not only in their binding operation, but also in their emphasis. While the HRR scheme aims to find a "reduced description" (see Hinton (1990)) to represent compositional structure, tensor-product binding aims to find a general connectionist representation for symbolic structure that allows storage of multiple symbolic structures in parallel and also provides a principled way of analysing interference between these structures.

When we describe the extended model below, we will see that the symbolic structure that we wish to represent does not pose a significant problem with expanding dimensionality. On the other hand, we do want to reduce the amount of interference between different bindings for our experiments. This makes the tensor-product representation suitable for our task.

In contrast, Holographic Reduced Representations have a serious limitation when it comes to noiseless retrieval of a component from a binding. We saw above that HRRs provide the capability to store multiple vectors by summing the individual associations. This summation means that patterns overlap each other in certain regions of the vector, leading to a distortion of the signal during retrieval. We saw that retrieval can be performed by circular cross-correlation operation, $\star$, but this operation leads to an *approximate* rather than exact unbinding. This approximation means that the result of unbinding contains some noise. Plate (1995) showed that this noise can be minimized under the constraint that each of the patterns are independently and identically distributed with zero mean and variance $1/n$, where n is the length of the vector. For example, if each element of each vector is randomly generated from a Gaussian distribution $\mathcal{N}(0, 1/n)$, then it can be shown (see Plate (1995)) that the noise will tend towards zero as the length of the vector increases.

However, this constraint makes the implementation of HRR binding complicated within our framework. We will see below that the vectors that need to be associated are generated by lexical, syntactic and semantic memories which generate sparse vectors with some nodes having high activation and most nodes having low or zero activation. These vectors do not obey the above constraint and therefore lead to interference between memories and a noisy retrieval. Plate (1995) did forecast this problem and suggested that it can be solved by mapping the vector generated by one's (memory) system onto a random vector (obeying the above constraints) through a hetro-associative memory. However, performing this additional mapping is not straightforward as it would require us to first train this hetro-associative network and then plugging in this trained network into our system. In addition, the presence of the additional step of a hetro-associative network will slow down the processing and, to our knowledge, lacks physiological justification.

These problems can be avoided by using the tensor-product binding mechanism. Though this mechanism does have the problem of expanding dimensionality, in our case it is not a severe restriction since we do not intend to develop a system with complex hierarchical symbolic structures.

§ 5.2.2 Extending the learning algorithm

In Section 5.2.1 we chose a mechanism for encoding associations of distributed representations. In this section we establish a learning mechanism for such representations.

Model ③ in the previous chapter had two independent learning mechanisms – short-term learning through hysteresis and a long-term learning through incremental adjustment to input sensitivity. But these learning mechanisms have a few pitfalls. Firstly, we have two contrasting learning mechanisms for explaining short term effects of priming and cumulative effects of priming. Although, Model ③ shows that both these mechanisms can work in conjunction to explain experimental findings, it would be preferable to have one uniform learning mechanism that is able to explain both short term and cumulative priming. Such a unified mechanism of learning would make it possible to directly compare the relative persistence of different kinds of memory and see if priming over different intervals can be explained as a consequence of these relative rates of persistence.

Secondly, we do not know how the long term learning mechanism presented in the last chapter *converges* over different regimes of learning. The reader would recall that when a node in Model ③ wins the competition, its input sensitivity increases by a fixed value. This means if a particular node keeps winning the competition, then its input sensitivity would keep growing. As the input sensitivity grows, the node would become even more likely to win the competition, leading to a self-perpetuating cycle of growth in input sensitivity. Physiologically this phenomenon is implausible. This raises the problem of ensuring convergence. In fact, unsupervised learning algorithms such as Hebbian learning face a similar problem and solutions to the problem (e.g., see Oja, 1982) involve establishing an explicit or implicit bound for the learning rule. In this section we would like to replace the learning rule with one that has more well established properties of convergence.

Lastly, we do not know how the learning mechanism *scales* when we use more than two symbolic structures. In Model ③, we used only two symbolic structures – represented by a PO and a DO node and a symmetrical set of inputs between the nodes. As the network size increases – i.e. we have more than two nodes – and the connection between the nodes becomes asymmetric, we do not know whether the network will converge to a stable activation pattern. Even though we will still be sticking to just two symbolic structures in the syntax layer, we will introduce a larger number of symbolic structures at the other layers and we would like our learning algorithm to have well

known convergence properties for representing more than two structures.

To sum up, we would like to replace the existing learning algorithm with an unsupervised learning algorithm for distributed representations that is suitable for memorising both short term and cumulative episodes, has well known properties of convergence and scalability and is consistent with our framework of dynamical systems. Auto-associative memory models, such as the Hopfield model, have well documented learning algorithms that meet these constraints (Hertz, Krogh, & Palmer, 1991). H. R. Wilson (1999) presented a dynamical system that extends such an auto-associative memory and uses a Hebbian algorithm to perform learning. We adapt this system to implement a learning algorithm that meets the above criteria and can be used in our system for encoding lexical, syntactic and semantic representations.

§ 5.2.2.1. **A network of excitatory and inhibitory nodes.—** The dynamical systems considered up to now have consisted of two nodes that are connected to each other through symmetric connections. We saw that such a system is capable of showing saddle-node bifurcations and these saddle-node bifurcations allow the network to behave in two different ways depending on the history of the system. Now, we generalise the idea to a set of N nodes, where N can be any number larger than one. Let us assume that these nodes are connected to each other through symmetric, excitatory connections. In addition, the network contains a global inhibitory mechanism that inhibits all the nodes based on the overall level of excitation in the network. We can formally describe this system with a set of two differential equations:

$$\begin{aligned} \frac{dE_i}{dt} &= \frac{1}{\tau}\left(-E_i + S\left(\sum_{j=1}^{N} w_{ij}E_j - 0.1G\right)\right) \\ \frac{dG}{dt} &= \frac{1}{\tau}\left(-G + g\sum_{i=1}^{N} E_i\right) \end{aligned} \tag{5.2.2}$$

where E_i is the activation of node i, G is the level of inhibition, τ is the time-constant for change in activation (or inhibition), S is the Naka-Rushton function (first seen on page 101) and w_{ij} is the connection strength (or synaptic weight) between node i and node j. The level of inhibition, G, can be seen as the level of activation of an inhibitory node. The crucial distinction between this node and all other nodes is that input from this node to any other node is inhibitory while the input from any other is always excitatory. If G is seen as an inhibitory node, then g can be seen as the value for the input synaptic weights of this node.

The above dynamical system extends the two-node system presented in the previ-

ous chapter in three ways:

1. Since the network can consist of more than two nodes, we can represent symbolic structure in the network using distributed representations. In the previous models, each node stood for a symbolic structure (a PO or a DO). This isomorphism between symbols and nodes does not need to hold in this multi-node network. We can now represent a particular symbolic structure as a pattern of activity distributed over a subset of nodes, which is the essence of distributed representations (Hinton, McClelland, & Rumelhart, 1986). Thus the dynamical system presented in Equation 5.2.2 allows us to move from localized to distributed representations.

2. In the previous models all connections were of the same type – either excitatory or inhibitory. The WTA system (on page 112) had inhibitory connections from each node to the other, while the STM system (on page 128) had excitatory connections. The system in equation 5.2.2 has both kinds of connections. The restriction lies in the type of node: the E node sends out only excitatory connections to all other nodes and the G node sends out inhibitory connections to all other nodes.

3. Models ①, ② and ③ held the strength of connections between nodes as constant. Equation 5.2.2 replaces these fixed connection strengths with variable synaptic weights, w_{ij}. This variation in connection strengths between nodes holds the key to learning. Instead of varying the sensitivity of a node to input, we vary these synaptic strengths. When a particular pattern of activation is presented to the network, the strength of connections between all activated nodes increases by a small amount. This is simple Hebbian learning. H. R. Wilson (1999) implemented the following modification of the Hebbian learning rule:

$$w_{ij} = k \cdot \mathcal{H}(E_i - 0.5M) \cdot \mathcal{H}(E_j - 0.5M) \tag{5.2.3}$$

where k is the fixed value of the connection that has undergone learning, M is maximum activation of a node and $\mathcal{H}(x)$ is the Heaviside step function, used to threshold the input:

$$\mathcal{H}(x) = \begin{cases} 1 & for \quad x > 0 \\ 0 & for \quad x \leq 0 \end{cases}$$

Therefore, the dynamical system presented in Equation 5.2.2 allows us to represent symbolic structure as a pattern of activity distributed over a subset of nodes. This pattern of activity can then be consolidated in memory using a modification of unsupervised Hebbian learning implemented by equation 5.2.3. Thus this network of inhibitory and excitatory nodes goes some way towards finding our desired learning mechanism.

What we require next is a way to find whether the learning algorithm converges. In other words, we can ask whether the learning algorithm in equation 5.2.3 leads to a stable solution for the dynamical system 5.2.2. We saw that the stability of the two-node system presented in the previous chapters can be established by comparing it to the additive-neuron model which has three points of equilibrium – two stable nodes and one saddle point. In a similar manner, we would like to establish whether the dynamical system in Equation 5.2.2 has stable nodes or saddle points. The existence of stable nodes will tell us that the network converges towards this node and the pattern of activity at the stable node will be the memory retrieved by the dynamical system. In addition to this, we also need to know the range of initial conditions that will move the network along a trajectory towards an asymptotically stable state. In other words, we want to know which initial patterns will cause the dynamical system to retrieve a particular memory. This range of initial conditions that converge to a stable equilibrium are called the *domain of attraction* of this stable solution.

The problem of determining the stable solutions of a dynamical system and the range of initial conditions that lead to these solutions was first investigated by Alexandr Lyapunov[3], who sought to establish the nature of trajectories for a dynamical system, without actually having to find the trajectories. The Lyapunov theory requires one to determine a state function, called the *Lyapunov function*, for the system. This function can be seen as a generalised energy function that plots the energy of the system in state space. As the system evolves, energy gets dissipated and the system moves in a direction to minimise this energy. Correspondingly, the Lyapunov function establishes a region around an equilibrium point such that all trajectories entering this region asymptotically converge towards the equilibrium (where the equilibrium corresponds to the point of minimum energy in the energy parlance). If one succeeds in finding a Lyapunov function around an equilibrium, it guarantees that the equilibrium point is stable. Thus, the procedure of finding the Lyapunov function helps one to establish whether an equilibrium is stable and also helps one find a range of initial conditions

[3]Russian mathematician, 1857–1918

(the stable node's domain of attraction) that will move the system along a trajectory towards this stable solution[4].

H. R. Wilson (1999) showed that it is possible to find a Lyapunov function for the dynamical system given by Equation 5.2.2 which learns via the unsupervised learning mechanism given in Equation 5.2.3. The existence of this Lyapunov function confirms that this dynamical system converges to a stable solution in such a way that a fraction of nodes are activated above threshold and the rest of the nodes have zero or very low activity. This is good news as it means that the system does not get into a self-perpetuating cycle where some active nodes end up activating all nodes above threshold and thus giving a meaningless result to memory retrieval.

H. R. Wilson (1999) explained that it is a balance between excitatory and inhibitory nodes that helps the network to achieve a stable solution with only a fraction of nodes firing. This analysis also helps us to uncover a limitation of this dynamical system. Since stability rests on the crucial balance between excitation and inhibition, this system has the limitation that the fraction of active nodes in a pattern are fixed – i.e. the number of active nodes in each pattern stored in memory needs to be the same. If the number of active nodes is (much) larger than this fraction, then excitation will dominate and, conversely, if only a small fraction of nodes are active, then the inhibition will dominate, driving all nodes to the rest state. In each case, the result of the retrieval will be a false memory. However, this is not a serious limitation as we will be considering a small number of patterns to be stored. Each of these patterns is generated systematically through the same procedure, ensuring that all patterns for a particular memory have the same number of active nodes.

Thus the dynamical system given by Equation 5.2.2 that learns via a Hebbian learning rule provides us with a memory mechanism for distributed representations that has well established analytical properties. When we discuss the computational details of the model below, we will modify the learning algorithm slightly so that it can be used for learning both short term and cumulative episodes of priming.

§ 5.2.3 Semantic extension

At the beginning of this chapter we set out to extend previous models so that the system is more plausible and more *complete*. So far we have considered theoretical extensions that will enable us to implement distributed representations and an analysable mech-

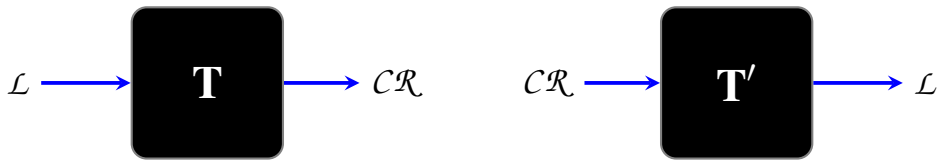
[4]Strictly speaking, it is possible to find multiple Lyapunov functions for a stable equilibrium. Thus the Lyapunov function establishes a sub-region of the complete domain of attraction.

anism for learning distributed patterns of activity. Both these extensions improve the plausibility of the system. In this section, we pursue the second goal and seek the theoretical apparatus required for extending the system so that it encodes processes of comprehension and production. By extending the system in this manner, we can then study how comprehension and production overlap and how learning during comprehension can lead to syntactic priming during production. Let us look in a bit more detail at what we mean by extending the system to include processes of comprehension and production.

First let us review how we have modelled comprehension and production until now. Comprehension was modelled by providing a large external input to one of the combinatorial nodes. Providing this large external input is equivalent of forcing the network to choose this node. When we simulate comprehension we try to model the cognitive processes of the listener. Since the linguistic choices are made by someone else (the speaker), our implementation simply forces the speaker to choose the grammatical category that has been used by the speaker. In contrast, during production, we want to simulate the speaker, who makes the linguistic choice. We implement this by allowing the model to choose a combinatorial node based on its internal state and the dynamics. We provide an equal amount of external input to both combinatorial nodes, making the decisions dependent on the system's hysteresis.

Now, we intend to extend these processes and make it similar to how information is processed by the cognitive system during comprehension and production. Instead of stating that comprehension requires providing a large external input for one of the nodes, we would like to treat the system as a black box and provide it only with an input signal. The system should simulate the lexical and syntactic layers based on this input and transform the input into some kind of a *canonical representation*. We will consider a mental representation to be canonical if different utterances with the same meaning map onto this mental representation. During production, we would like to get the reverse transformation: the system is given this canonical representation and should turn it into an output signal. We will assume that all representations up to the semantic level are canonical, focussing the burden of choice between different utterances at the syntactic level. In other words, when the system faces a choice between two utterances (during production) we will assume that this choice is completely made at the syntactic level because the utterances are identical (canonical) at the semantic (or higher) level.

Ideally, the input signal for the comprehension and the output signal for production would be an acoustic signal. At the other end, the canonical representation should be



Comprehension Production

Figure 5.2.2: [Black box] Comprehension transforms linguistic signal ($\mathcal{L}$) into a canonical representation ($\mathcal{CR}$). The transformation involves flow of information through lexical and syntactic representations. Production involves the opposite mapping – from a canonical representation to a linguistic signal.

something like an intentional state. However, this might be a very ambitious project bridging philosophy of mind to speech technology and crossing, in between, the landscape of computational linguistics and psychology.

We would like to limit ourselves to a more modest goal – one that is relevant to the study of syntactic priming. In this section we consider the nature of the canonical representation and in the next we would look at the input representation. We will see that the input representation is restricted to a text tagged with the syntactic function of each word. But we will not get too far ahead of ourselves just now and first concentrate on finding a reasonable definition for the canonical representation. This canonical representation will form the output of a comprehension trial. Because we intend to extend our framework to study the effect of semantic structure on syntactic priming, we restrict the canonical representation to be the semantic structure of an utterance.

What is the nature of such a canonical (semantic) representation? Let us illustrate the constraints on such a representation with the help of an example. Consider the utterance *John gave Mary the book*. At the syntactic level, we can say that *John* is the subject of the sentence with two objects, *Mary* and *book*, and the verb *gave*. When we move to the semantic level, the subject *John* needs to be associated with the concept JOHN[5], just as *gave, Mary* and *book* need to be associated with the concepts GIVE, MARY and BOOK. Furthermore, at the syntactic level, *John, gave, Mary* and *book* are bound together by their syntactic roles in the utterance. Similarly, at the semantic level, the concepts JOHN, GIVE, MARY and BOOK need to be bound to each other in such a way that the semantics of this binding is associated with the meaning of the whole utterance. If we represent the semantics of the entire utterance with the predicate-

[5]We will consistently use UPPERCASE (sans-serif) letters to signify a conceptual element, *Italics* (roman) to refer to word forms or syntactic functions and **BOLD** (monospace) to refer to predicates.

argument structure **GIVE**(JOHN,MARY,BOOK), then this predicate-argument structure needs to be associated with the entire utterance *John gave Mary the book*. Lastly, this semantic representation needs to be canonical – i.e. different (valid) rearrangements of the words should lead to the same semantics. The prepositional dative *John gave the book to Mary* and the double object dative above should both map to the same semantic representation.

This example shows multiple levels of associations between syntactic and semantic representations. At one level, the word forms are associated with lexical concepts (e.g.*John* and JOHN) and at another level the entire semantic structure is associated with the syntactic structure (e.g.**GIVE**(JOHN,MARY,BOOK) and *John gave Mary the book*). We also see that the verb plays a special role as it picks out the predicate which provides a list of arguments. These arguments are matched to the concepts picked by the subject and object of the utterance. These different associations between syntactic and semantic structures are schematically shown in figure 5.2.3.

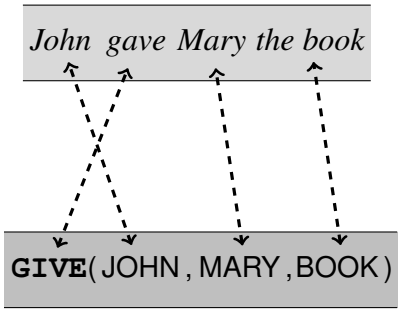


Figure 5.2.3: [Dative mapping] This figure shows the mapping between the syntactic elements of the construction *John gave Mary the book* and the semantic components of the predicate-argument structure **GIVE** (JOHN, MARY, BOOK).

We can now specify a semantic representation based on these constraints. The semantic layer needs to perform two different kinds of functions. Firstly, it needs to represent the individual lexical concepts and secondly, it needs to represent the binding between different concepts. These two functions can be used to divide the semantic representation into two types of components: *lexical concepts* and *schemata*. Each word in an utterance picks out a lexical concept in the semantic layer. This lexical concept represents the semantic knowledge related to the word form. In addition, each *verb* picks a schema which evokes a body of knowledge used for understanding the relationships between the lexical concepts. The idea of schemata has been used in connectionist networks for organising conceptual knowledge (see, for example

(Rumelhart et al., 1986)) and is closely related to the idea of *frames* used in cognitive linguistics (Fillmore, 1985). The schema or frame, picked by the verb, exposes a set of roles which are matched with corresponding lexical concepts picked by the utterance. Together, the frame along with the matched lexical concepts represents the semantics of the entire syntactic construction. Figure 5.2.4 shows an example of a frame picked by the verb *risk*. It also shows how the sentence *You've risked your health for a few cheap thrills* provides lexical concepts that serve as arguments to the RISK frame and link to its roles.

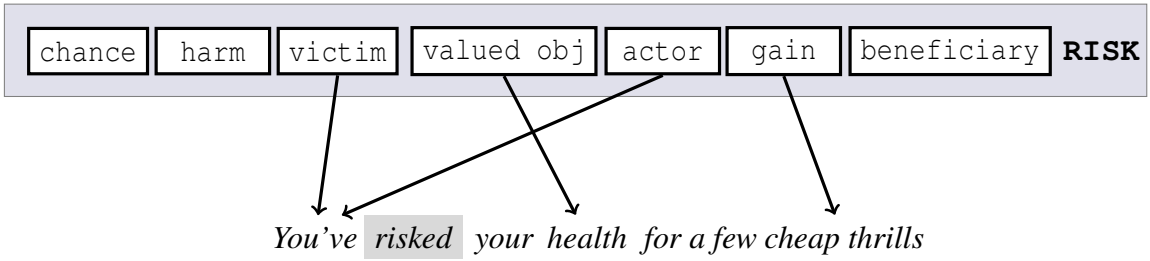


Figure 5.2.4: [Risk frame] The analysis of the arguments (roles) of the **RISK** frame and how different concepts in the utterance *You've risked your health for a few cheap thrills* attach to these roles. Note that the frame contains a larger body of knowledge and an utterance activates only part of this knowledge. (Based on the analysis of RISK frame given in Fillmore and Atkins (1992))

§ 5.2.3.1. **Concepts, Schemata and construction grammar.—** This representational scheme is evocative of a popular grammatical theory. In our scheme, the syntactic layer consists of word forms and utterances; the semantic layer consists of lexical concepts and frames. The part-whole relationships between words and utterances at the syntax layer are reflected in the relationships between concepts and frames in the semantic layer. The parts of a syntactic structure (i.e. the utterance) map to the parts of the semantic structure (i.e. the frame). At the same time, the whole syntactic structure also maps onto the whole semantic structure. Croft and Cruse (2004) pointed out that such pairing of word form and meaning is a key property of construction grammars such as Cognitive Grammar (Langacker, 1987) and Construction Grammar[6] (Fillmore & Kay, 1993). Because of this similarity in our goals of semantic representation and construction grammars, we would like to investigate what this formalism can tell us

[6]There is a bit of confusing terminology here, with Construction Grammar (capital C and G), proposed by Fillmore and Kay (1993), being one type of construction grammar.

about the relationships between syntactic and semantic structure and whether a description consisting of the conceptual content of words and their role in a schema provides a canonical representation for utterances.

Construction grammars treat any syntactic configuration as a construction. The parts of the syntactic configuration might be entirely variable, like in our double object dative *John gave Mary the book*, or they may be fixed, like the utterance *What's this cat doing in here?*, where the parts *What's* and *doing* are fixed (Kay & Fillmore, 1999). Sentences such as *What is this scratch doing on the table?* and *What am I doing reading this paper?* are illustrations of other instances of this construction. Kay and Fillmore (1999) argued that all constructions of the form *What's X doing Y* share some semantic features and that the entire semantics of such an utterance cannot be construed from the semantics of its parts. Thus representation of the semantics of such utterances presents a problem. This problem is solved by construction grammar which takes a construction as an atomic unit that cuts across phonological, syntactic and semantic levels and associates each construction with its own semantic interpretation. Thus construction grammars do not just map word forms with lexical concepts, they also map entire syntactic configuration onto a semantic structure.

Croft and Cruse (2004) illustrated this mapping between different elements of the syntactic and semantic structures using the example of an intransitive utterance such as *Heather sings*. The construction grammar representation of this utterance is shown in Figure 5.2.5. The word forms *Heather* and *sings* are (vertically) linked to the corresponding lexical concepts HEATHER and SING. In addition, the whole intransitive construction *Heather sings* is also linked to the whole semantic structure SING (HEATHER). The figure also shows the (horizontal) relationship between the syntactic elements *Heather* and *sings* – i.e. the subject-predicate relationship. In the same way, HEATHER and SING are related to each other through the predicate-argument relationship.

Similarly, our scheme for semantic representation can be seen as mapping both the individual word forms such as *John* and entire syntactic structure such as the double-object dative, onto their semantic counterparts – lexical concepts and schemata. Thus the schema, with its relevant roles related to other semantic components, constitutes the semantic structure that corresponds to the entire construction.

Construction grammar also helps us understand the different kinds of relationships between syntactic elements and semantic components. Just like HEATHER and SING were connected through a predicate-argument relationship in the above example, our

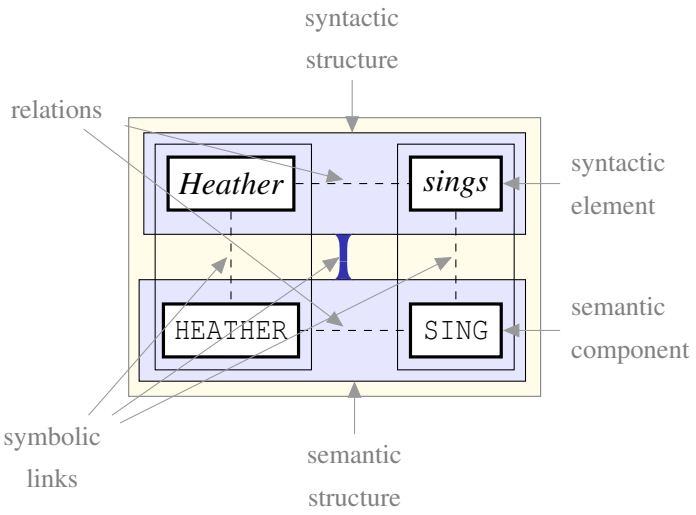


Figure 5.2.5: [Heather sings] A construction grammar representation of the intransitive sentence *Heather sings*. Horizontal dotted lines show *relations* (syntactic and semantic) while vertical dotted lines show *links*. The vertical bold line shows the link between the entire syntactic structure – the intransitive construction – and the entire semantic structure – the predicate-argument construction. (Adapted from Croft and Cruse (2004)).

example connects **GIVE** and JOHN through a similar predicate-argument relationship. In a double object dative, the verb **GIVE** forms the predicate, because it is *relational* – i.e. it presupposes another concept (Croft & Cruse, 2004). On the other hand the subject (JOHN) does not presuppose another concept and therefore forms a filler. The frame contributes a list of roles and the fillers link to these roles. In this manner the different semantic components in an utterance connect to each other forming a cohesive whole.

This comparison with construction grammar also highlights an assumption of our representational scheme. We have assumed that verbs always contribute the frame containing the roles while subjects and objects contribute the fillers that attach to these roles. This special status of the verb in performing the functional assignment during language production is not unique to our account and has some empirical justification. Bock and Cutting (1992) showed that people make fewer agreement errors if utterances are hierarchically organised, guided by requirements of verbs. Based on this data, Bock and Levelt (1994) argued that verbs supply the argument structure that controls the function assignment within a clause.

While this special status of verbs is simple to understand for datives (or transitives),

which we intend to use for our simulations, construction grammar helps us see how this scheme can be extended for more complex sentences. Croft and Cruse (2004) pointed out that in an utterance such as *You should read this article*, the verb **READ** is simultaneously a predicate and an argument. Our simple scheme does not have the capability of representing such sentences, but can easily be extended using the theory of Cognitive Grammar which allows a frame to become a filler in another frame (Langacker, 1987).

There is another kind of relationship that is highlighted by construction grammar. This is the relationship between different syntactic elements and the entire syntactic construction. In the prepositional dative *John gave the book to Mary*, the words *John*, *book* and *Mary* are nouns, but they bear different relationships to prepositional dative construction. *John* serves the function of the subject, while *book* and *Mary* are the first and second objects of the construction. Croft and Cruse (2004) pointed out that this is the *part-whole*, or *meronomic*, relationship between each element and the whole construction and contrasts with the syntactic relation between one element of the construction – say, the subject – and another element – say, the verb. In our model, we will stress the part-whole relationship and we will assume that this knowledge resides within the grammar (a part that we do not implement).

Lastly, this frame-based semantic structure satisfies the requirement for being canonical. The semantic structure for a prepositional dative consists of a frame with its roles linked to the lexical concepts for the subject and objects of the utterance. Suppose that these roles are `P`, `Q` and `R` and the lexical concepts are *john*, *mary* and *book*. Then the semantic representation consists of the frame, **VERB**(`P` → *john*, `Q` → *mary*, `R` → *book*). This representation will be canonical if the double-object dative has the same semantic representation. Indeed, the semantic representation does not contain any elements that will be different for the two structures. The syntactic relationships within the word forms should ensure that the roles `P`, `Q` and `R` are mapped to the same lexical concepts, making the semantic representations for the PO and DO dative identical.

We have proposed a scheme of representing the semantic structure of an utterance through mapping the syntactic elements into two kinds of semantic components. The word forms are mapped onto lexical concepts, which can be implemented as patterns of activity in the semantic space; the entire syntactic construction is mapped onto a semantic structure which consists of a frame and contains roles that are, in turn, linked to the lexical concepts. We have seen that this scheme allows us to obtain a canonical representation which can be used as the output of a comprehension episode, or as the

input of a production episode. We have also seen that this scheme is loosely related to the formalisms of construction grammar, which allow us to ground this scheme into a well established framework and provide the capability of expanding this scheme into a more general representation for different types of linguistic utterances.

5.3 Computational Implementation

Equipped with the theoretical extensions presented in the last section, we are now ready to extend our model from the previous chapter into a framework for studying structural priming during comprehension and production. In this section, we will present the architecture of the extended model, discuss its formal implementation, introduce the processes required for implementing comprehension and production and see how these processes can lead to structural priming.

§ 5.3.1 Network architecture

Before we lay out the architecture of the extended model, we will motivate the representations of this model by looking at the kind of information processing it needs to do. In Section 5.2.3, we stated that our computational system provides an interface between a linguistic signal and a canonical representation. The linguistic signal lies in the external environment; it is either produced by the computational system or needs to be comprehended by it. The canonical representation lies inside the cognitive system; it is either the product of comprehension or the origin of production. Both the linguistic signal and the canonical representation are forms of information and as this information flows through the computational system, it undergoes a transformation. This transformation requires a number of intermediate steps, as information changes from one representation to the other.

Consider the familiar example of comprehending the double object dative *John gave Mary the book*. The input signal is the linguistic signal consisting of a sequence of words. The system needs to look up each word in its memory and match it to a lexical concept. Thus the system would search its memory for the word form *John*, retrieving the corresponding concept JOHN, then *Mary* and so forth. While retrieving these concepts, the system must somehow simultaneously also record what it retrieved – i.e. as a result of the retrieval, the system must itself undergo a change. Changing itself is important for the system because we are studying priming and we want to see

how processing one utterance affects subsequent processing. The system must also retrieve the grammatical structure for the utterance. We shall assume that the system is given the syntactic function of each word. So the system must use this information to retrieve, from its memory, the fact that the given sentence is a double object dative. Again, as the system retrieves this information, it should undergo further learning so that this episode of comprehension influences future episodes. Once the system has retrieved both the conceptual information and the structural information corresponding to the input sentence, it must use this information to generate the required canonical representation. We saw that this canonical representation consists of a schema along with the bindings between the roles inherent in the schema and the concepts retrieved by the system. Thus the system must retrieve a schema (corresponding to the relational concept, **GIVE**) from the memory. This schema will present a list of arguments which will form the roles that need to be attached to the concepts that were previously retrieved. The system should then use the grammatical structure that it retrieved earlier to see how the roles should attach to the concepts and perform this binding. The binding of the schema and the lexical concepts forms the canonical representation that was desired.

This example shows that the system needs to access three different kinds of memories: (i) a memory for lexical concepts, (ii) a memory for the structure of the utterance and (iii) a memory for the schema for the entire construction. The memory for lexical concepts and schemata is retrieved based on the input word forms and the memory for structure is retrieved based on the input parts-of-speech. While we consider an example of language comprehension, the same memory types can be used to perform the reverse mapping from a canonical representation to a sequence of words during production. Thus a chief architectural change from Model ③ (presented on page 138) is the inclusion of three kinds of representation layers – one for each of the above kinds of memory – in place of the lexical and syntactic layers in Model ③.

Figure 5.3.1 shows the network architecture of the extended model. The three kinds of memory are represented as three different representational layers. Each layer is connected to both the other layers through a set of associative links. These associative links are similar to the links between the lexical layer and syntactic layer in Model ③ and serve as the short term memory of an utterance. The three representational layers themselves retain long term syntactic, conceptual and schema memories. This model does not implement the memory for the word forms themselves. Instead it collapses the representation for word forms and lexical concepts, assuming each word

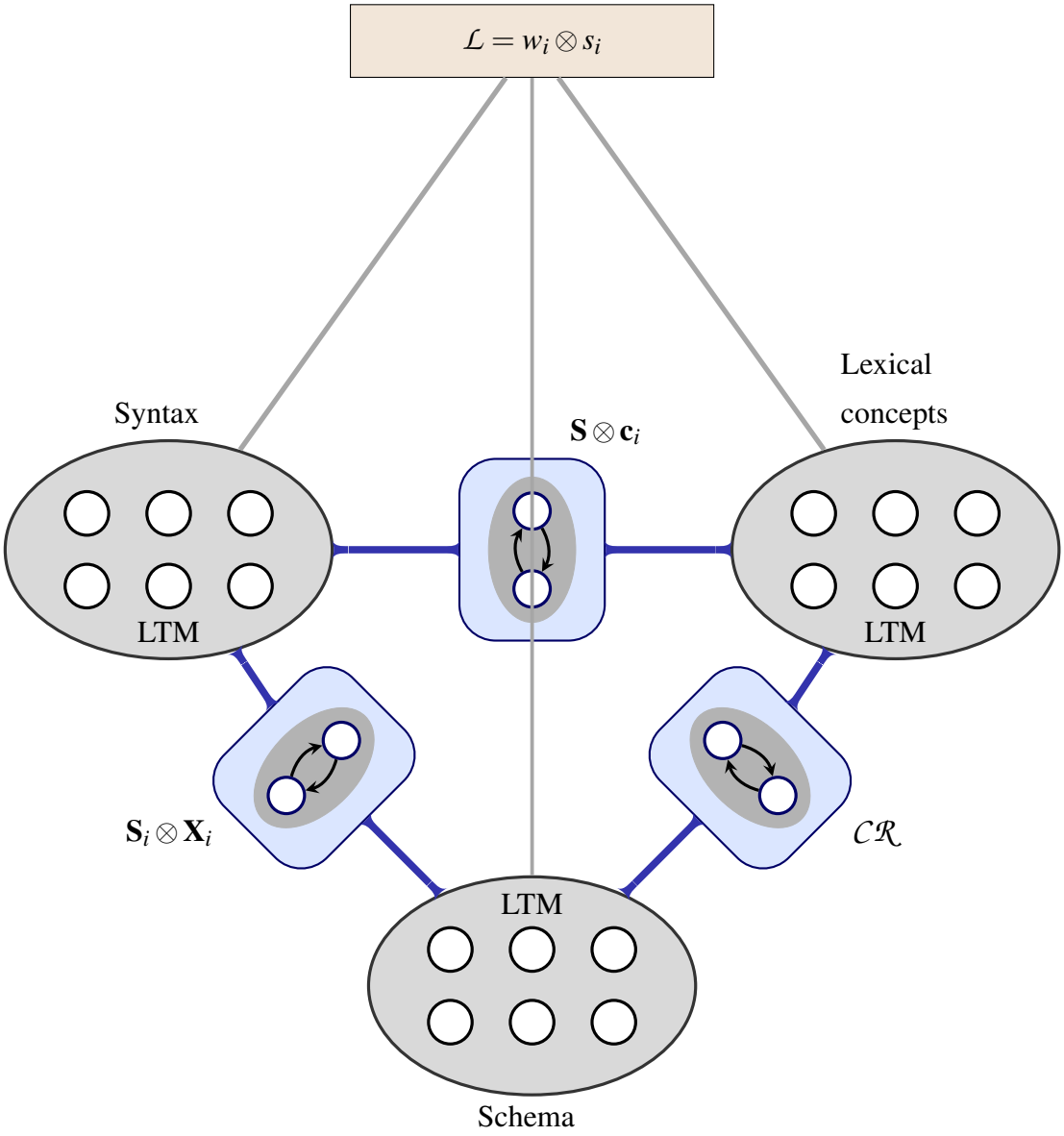


Figure 5.3.1: [Extended model] The system consists of three long term memory (LTM) modules that represent syntax, lexical concepts and schema, through a distributed representation scheme. Each pair of long term memories is associated with tensor-product based short term memory modules. During each episode of comprehension or production, the system performs a transformation between a linguistic signal $\mathcal{L}$ and a canonical representation $\mathcal{CR}$.

form corresponds to one lexical concept. A similar simplifying assumption is made by other models of language processing, such as Roelofs (1992, 1993).

It is instructive to understand this extended model from the perspective of information processing during comprehension– i.e. how information gets transformed when it passes from the environmental domain to the cognitive domain modelled by the network. The linguistic signal, which we take to be a sequence of (syntactic-function marked) words, is analysed into three different modules as it passes through the system. Each of these modules matches a particular property of the signal with an internal long term memory of such properties. The syntax layer, for example, tries to match the syntax of the linguistic signal with a long term memory of such a syntax. Thus the model suggests that the process of understanding a linguistic signal involves an analysing (or transforming) the linguistic signal into these three kinds of information. Once this analysis has been achieved, the analysed information is then combined through the associative links. The integration of the different forms of information is the model's memory of a linguistic *episode*. The memory of the surface form of the episode is retained in the association between the syntactic and lexical layers (or here, the lexical concepts layer). The memory of the content (or the semantics) of the episode is retained in the association between the conceptual and schema layers. This memory is nothing but the canonical representation of the linguistic signal. Thus, from an information processing perspective, we can say that the role of the three modules of long term memory is the *analysis* of information just as the role of the three associative modules of short term memory is its *integration*.

The question now arises of how is a sequence of words analysed into these three kinds of representations, or rather, who does this analysis. How can a sequence of words be converted into a pattern of activity over syntactic, conceptual and schema memories? There has to be a cognitive procedure that can convert such a signal into each of these cognitive representations and vice-versa. This cognitive procedure will have to learn how to perform this mapping and therefore it forms a developmental account of learning language. Chang et al. (2006), for example, provided such a developmental account that maps a sequence of words into an internal representation for the syntax. This study does not intend to build such a developmental account and instead assumes that this procedural knowledge is available in a given module. As we did in the previous models, we concentrate on an adult subject who has already acquired this procedural knowledge and is able to perform this mapping.

We make the distinction between a procedural learning that is required to map the

linguistic signal into each of these representations and episodic learning that remembers each of these representations once they have been invoked. While the procedural knowledge is acquired over a period of time through the repeated presentation of data, episodic knowledge is acquired as a result of a single presentation. Our extended model performs such episodic learning and we are interested in investigating whether this form of learning can explain patterns of structural priming.

§ 5.3.2 Formal Description

This section presents a formal description of the model's properties. We describe how the model represents the external input, how it represents internal information and how it learns this information. We begin by describing the dynamics of the model, which are closely related to the model's learning. Then we discuss how the model forms its internal representations and we describe some specific representations. Lastly, we discuss how the model represents the input. Since we are studying both comprehension and production, we deal with two kinds of inputs – the input during comprehension (i.e. the linguistic signal) and the input during production (i.e. the canonical representation). It is needless to say that this automatically ensures that we also represent the output.

The model consists of two kinds of memory modules – the long term memory modules, which store the syntactic, conceptual and schema representations, and the short term memory modules, which store the associations between these modules. We call the former *long term* memory because this form of memory stores the long term knowledge of syntactic constructions, lexical semantics and relational properties of utterances. In contrast, the *short term* memory stores the transitory knowledge of how each of these representations is associated in a particular episode of comprehension or production. These two kinds of memories are implemented by different dynamical and learning principles and we discuss each one in succession.

§ 5.3.2.1. **Dynamics and Learning of LTM.—** Each of the long term memory modules is implemented using the network of excitatory and inhibitory nodes presented in Section 5.2.2.0. The dynamics of this system are given by the Equation 5.2.2 and the system learns the weights between nodes w_{ij} via an unsupervised learning mechanism. We propose a learning algorithm that extends the Hebbian learning mechanism presented in Equation 5.2.3:

$$w_{ij} = k \cdot \mathcal{H}(E_i - 0.5M) \cdot \mathcal{H}(E_j - 0.5M)$$

This learning algorithm sets the weights between two active units to a constant value k. H. R. Wilson (1999) used this mechanism to perform a single-shot (episodic) learning of an input pattern. But since we are studying structural priming, we want this learning algorithm to explain not just single-shot learning, but also how the memory of linguistic representations changes during a discourse. We would like to explain (a) how an episode of priming consolidates a pre-existing memory, and (b) how memory shows a bias towards representations that have been activated more recently. To explain these two phenomena, we change the Hebbian learning algorithm of Equation 5.2.3 in three different ways:

- CUMULATIVE LEARNING. Experiments such as Kaschak and Borreggine (2008) and Hartsuiker and Westenberg (2000) show that priming accumulates over a series of priming trials. These results cannot be explained by equation 5.2.3 (page 205) because this equation changes the weights after a single trial to a constant value, k. Subsequent presentation of the same pattern will have no effect as the equation would still keep the value of the weights constant at k. Instead, we would like priming to accumulate over a series of trials. As we saw in the last chapter, where we adjusted the input sensitivity, this cumulative effect can be achieved through incremental learning – i.e. each episode increases the weights by a small amount λ:

  $$w_{ij}^{T+1} = w_{ij}^{T} + \lambda \cdot \mathcal{H}(E_i - 0.5M) \cdot \mathcal{H}(E_j - 0.5M)$$

  where w_{ij}^{T} is the connection weight between nodes i and j after trial T. This equation says that if two nodes are activated during a trial, then their connection weight will undergo a small increment λ, otherwise it will remain the same.

- BOUNDED WEIGHTS. The modification made for cumulative learning presents the problem of unbounded growth of weights. When a pattern is repetitively presented, the connection strengths between active nodes will grow after each trial in an unbounded manner. This unbounded growth of weights is a problem from two perspectives. Firstly, it is physiologically implausible as synaptic weights vary only within a limited range of values (Senn & Fusi, 2005). Secondly, unbounded weights in our excitatory and inhibitory network (equation 5.2.2) will disturb the balance between excitation and inhibition causing excitation to dominate. This dominant excitation would mean that activation would spread to the entire network and the network will no longer display the stable equilibriums

required to encode memories. Therefore, in order to prevent this unbounded growth of weights, we change the learning algorithm so that it stops to increase weights above a particular bound:

$$w_{ij}^{T+1} = min(w_{ij}^{T} + \lambda \cdot \mathcal{H}(E_i - 0.5M) \cdot \mathcal{H}(E_j - 0.5M),\ W_{max}) \tag{5.3.1}$$

where $min(\cdot)$ is a function such that $min(x,y)$ is equal to the minimum of x and y. This learning algorithm ensures that the connection weights do not increase above an upper bound W_{max}.

- RECENCY AND DECAY. Our last modification to the learning mechanism is to accommodate the fact that as compared to more remote episodes of language processing, recent episodes have a larger impact. This larger impact can be more precisely characterised in terms of the size of each pattern's domain of attraction. If one pattern is presented more recently than the other, then the domain of attraction of the more recent pattern should be larger. A larger domain of attraction ensures that the memory pertaining to the attractor will be retrieved for a larger number of initial conditions. The domain of attraction of a particular memory can be increased, in a convergent manner (Hertz et al., 1991), by increasing the connection weights through Hebbian learning. But the bounded Hebbian learning algorithm in equation 5.3.1 means that most memories will quickly reach the upper bound leaving no discrimination between recent and remote episodes. Therefore, we need an explicit mechanism that changes the weights so that remote memories have smaller weights as compared to recently activated memories. In other words, we would need to build in an explicit mechanism of decay in memory. Because of the same arguments for the decay mechanism in the previous chapter, we consider an exponential decay in memory:

$$w_{ij}^{t+\Delta t} = w_{ij}^{t} + \exp^{-\Delta t/\tau_{ltm}} \tag{5.3.2}$$

where $w_{ij}^{t+\Delta t}$ is the weight at time $t+\Delta t$ and τ_{ltm} is the time-constant for decay in long term memory. Note that we use t to refer to time, whereas we used T (uppercase) to refer to the trial in Equation 5.3.1. While learning occurs at the end of every trial (T), we assume decay occurs as a function of time (t). The reason is that we would like to control the amount of decay as a function of the number of fillers between prime and target and we will manipulate this distance between prime and target by changing the duration of fillers.

We also note that explicitly encoding decay is only one of the possible methods of accounting for recency. Sikstrom (1999, 2002) showed that a learning rule where weights are bounded and, crucially, the amount of learning depends on the weight itself will lead to exponential decay. We chose not to use this learning algorithm because we would like to dissociate learning regimes (priming trials) from forgetting regimes (filler trials). Our algorithm achieves this dissociation by the learning mechanism in equation 5.3.1 after priming trials and the decay mechanism in Equation 5.3.2 during filler trials.

§ 5.3.2.2. **Dynamics and Learning of STM.—** Each pair of long term memory modules are connected via a layer of short term memory. We will soon see that each long term memory module represents information as a pattern of activity distributed over the nodes. Thus each short term memory module needs to bind a pair of distributed representations. We saw above (Section 5.2.1.1) that such pair of distributed representations can be bound using the tensor product.

Though the three short term memory modules represent three different associations, they are implemented using the same scheme. Consider two kinds of memories S_i and X_i. For the sake of illustration, let us say that S_i is a syntactic memory and X_i is the memory of a schema. Now using the representational notation used above (originally in Smolensky (1990)), we can represent the connectionist representation (ψ) of these two memories as $\mathbf{S}_i$ and $\mathbf{X}_i$, where $\mathbf{S}_i = \psi(S_i)$ and $\mathbf{X}_i = \psi(X_i)$. The short term memory associating the syntax and schema representations can be found using the tensor product of the two connectionist representations:

$$\psi(S_i/X_i) = \mathbf{S}_i \otimes \mathbf{X}_i$$

The right hand side of this equation gives us a pattern (a tensor) of activations that can be stored in the short term memory. This brings us to the question of how the short term memory stores a given pattern of activation. More specifically, we would like to find out how the short term memory combines the activity patterns that it has previously stored with a new pattern. This is the problem of superimposing different bindings in a memory. We saw that both the Holographic Reduced Representations and the tensor-product binding superimpose activity patterns using mathematical summation. Smolensky (1990) showed (and we reviewed in section 5.2.1.1) that summation leads to memories that can be unbound using the inner product. We adopt this mechanism of superposition of patterns to store a new pattern in a given short term memory. Thus, for a set of n different syntax and schema patterns, we can calculate the short term

memory representation as:

$$\psi\left(\bigwedge_{i=1}^{n} S_i/X_i\right) = \sum_{i=1}^{n} \mathbf{S}_i \otimes \mathbf{X}_i$$

which says that the connectionist representation for the superposition of n associations S_i and X_i can be found using the sum of the tensor products of the connectionist representation for each association.

Lastly, just like in the long term memory case, we would like to establish how this short term memory changes with time. Our concept of short term memory as the activation-based association between two representations is the same as the binding layer of Model ② and ③ is the previous chapter. We saw that the binding nodes in Model ② and ③ remained active for a short period of time and then decayed catastrophically. We used this behaviour to explain the difference in the longevities of structural priming and lexical boost.

While implementing the decay characteristics of the short term memories in the extended model, we keep in mind this difference in longevities of the two kinds of memories. We can actually use the same computational implementation of the short term memory as used in Model ② and ③ – i.e. mutually excitatory nodes (Figure 4.4.2 on page 127). However, in order to avoid computational complexity, we implement the memory as a simple exponential decay:

$$a_i^{t+\Delta t} = a_i^t + \exp^{-\Delta t/\tau_{stm}} \tag{5.3.3}$$

where a_i is the activation of the i-th node and τ_{stm} is the rate of decay of the activation. We will set the value of τ_{stm} to be much smaller than τ_{ltm} above, making the rate of decay in short term memory much faster than that of long term memory. This fast decay effectively mimics the short term catastrophic decay in binding nodes that we saw in model ③.

§ 5.3.2.3. **Representations.—** Now that we have formally described the computational level of the model, let us shift our attentional to the representational level and see how the model represents different kinds of symbolic structures. The model deals with the following symbolic structures: the linguistic input ($\mathcal{L}$) which consists of the word forms (w_i) and the syntactic categories or *elements* (s_i) associated with these word forms; the whole syntactic constructions (S) which bear the part-whole relationship to its syntactic elements; the schema (X) invoked by a predicate and the roles (arg_i) that form the arguments of the schema; the concepts (c_i); and the canonical representation $\mathcal{CR}$ formed by associating roles with concepts.

Our goal is to specify a mapping ψ that takes each of these symbolic structures to a pattern that can be represented in a connectionist network. Following Smolensky (1990) we assume that this pattern is a vector of activity states of the connectionist network. Therefore ψ is a mapping from a *symbol space* to a *vector space*. We will also assume that the mapping ψ is such that the vector obtained is sparse and normalised. The sparseness of the vector ensures that the overlap between different patterns is minimised and the normalisation ensures that different signals have the same power.

First let us look at the syntax layer which needs to represent the syntactic construction S. This syntactic construction is composed of a list of syntactic elements s_i. We can represent each syntactic element as a vector $\mathbf{s}_i$, so that $\mathbf{s}_i = \psi(s_i)$. Once we have the connectionist representation for the syntactic elements, we can build the entire syntactic construction using these elements and a knowledge of what relation r_i each syntactic element bears to the whole syntactic construction. This step of assembling a syntactic construction from a list of syntactic elements and their (part-whole) relationship to the construction is equivalent to the step of 'constituent assembly' in traditional models of language production (Chapter 2, page 22). The only part-whole relationship we implement in our model is the position of different syntactic elements in the syntactic construction. For example, in the DO dative construction the indirect object of the verb comes immediately after the verb and this knowledge resides in the grammar, from where it is retrieved and encoded into the relation r_i. A more elaborate model can replace this method of associating syntactic elements directly to their positions by an incremental grammar that determines a hierarchical constituent structure and orders syntactic elements in different positions based on this structure. In our model, we built a syntactic construction by associating each syntactic element with its position and superimposing these associations. This operation of associating vectors is a familiar one and can be performed by the tensor-product binding:

$$\mathbf{S} = \sum \mathbf{s}_i \otimes \mathbf{r}_i$$

where $\mathbf{r}_i = \psi(r_i)$ and $\mathbf{S} = \psi(S)$ are the vectors representing the (positional) relation and the syntactic construction respectively.

The representational scheme of the syntax layer are quite similar to that of the schema layer. Where the syntax layer has syntactic elements, the schema layer has roles in the schema. Each role arg_i can be represented by the vector $\mathbf{arg}_i$, by performing the mapping $\mathbf{arg}_i = \psi(arg_i)$. Just like we assumed that different syntactic elements had a part-whole relationship with the syntactic construction, we also assume that the

different roles in a schema have a part-whole relationship in the schema. This part-whole relationship is encoded by pos_i, the position of the role in the schema. Again, our notion of this part-whole relationship is very simplistic and a grammatical formalism like Cognitive Grammar can be used to expand on this relationship. However, given this simplistic notion of the relation of a role to the whole schema, we can state the connectionist representation of the whole schema as:

$$\mathbf{X} = \sum \mathbf{arg}_i \otimes \mathbf{pos}_i$$

where $\mathbf{pos}_i = \psi(pos_i)$ and $\mathbf{X} = \psi(X)$.

Finally, the linguistic signal is an association between the list of words and syntactic elements, while the canonical representation is an association between the roles of a schema and concepts:

$$\mathcal{L} = \sum \mathbf{w}_i \otimes \mathbf{s}_i$$
$$\mathcal{CR} = \sum \mathbf{arg}_i \otimes \mathbf{c}_i$$

where $\mathbf{w}_i = \psi(w_i)$ and $\mathbf{c}_i = \psi(c_i)$ are the connectionist representations for a word and a concept respectively.

These representations provide the information structures that can be processed by the algorithms on our extended model. The representational scheme also clarifies where the system relies on a procedural knowledge that we assume to be available to the system at the outset. This is the knowledge encoded in the mapping ψ which maps a symbolic structure to a vector of activity states. The system also assumes the availability of grammatical knowledge encoded by the two structures, r_i and pos_i which encode the part-whole relationships for syntactic and schema constructions. As the system processes information during comprehension and production, it will manipulate the different information structures that we discussed above, but the procedural as well as the grammatical knowledge will stay the same.

§ 5.3.3 Comprehension and Production

Let us now look at how the network performs the procedures of comprehension and production. Both procedures can be seen as a flow of information through the different memory modules. After describing the step-by-step flow of information, we discuss the overlap between the two procedures and give and example to illustrate this overlap. In each direction of information processing, we will see that information is first analysed – i.e. broken down – and then integrated – i.e. combined.

§ 5.3.3.1. **Comprehension.—** The system comprehends by simulating each of the three memory modules in response to an input sequence of words and generates a canonical representation for this sequence as a result. We can describe comprehension as a transformation from a set of words and their syntactic functions ($\mathcal{L}$) to the canonical representation ($\mathcal{CR}$) through the following sequence of steps:

Step 1 Determine the concepts. Each input signal consists of a list of words. These words will give a word form vector, obtained by applying the mapping ψ. We also assume that each word form corresponds to a concept. Therefore, each word in the input signal provides a direct mapping onto a conceptual pattern.

Step 2 Determine the syntactic elements. The input signal gives a list of syntactic functions corresponding to each word. Again using the mapping ψ, obtain a vector for the connectionist representation for each syntactic element in the linguistic signal.

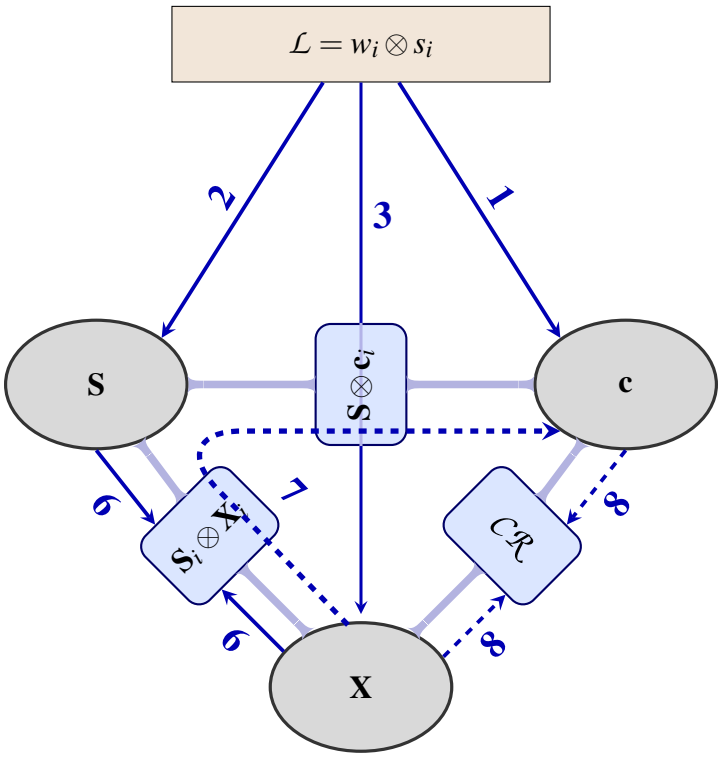


Figure 5.3.2: [Comprehension] The edges label the order in which information flows during comprehension. The *forward pass* is shown using bold lines while the *backward pass* is shown using the dashed line.

Step 3 Determine the schema. The word associated with the syntactic function *verb* will provide the key to determine this vector.

Step 4 Assemble syntactic elements into a syntactic construction. Calculate the syntactic pattern of the entire syntactic construction by binding the patterns for all the syntactic elements.

Step 5 Simulate the syntactic, conceptual and schema memories and learn the input patterns. Steps one to four provide the patterns required as cues for each memory module. Using these cues retrieve a memory. The process of retrieval will automatically lead to learning.

Step 6 Bind syntax and schema. Calculate the binding between syntactic and schema patterns and store this binding in the STM module associating the two layers.

Step 7 Determine a concept for each role. Determine the roles of the schema and use the binding between the schema and syntactic construction to determine the syntactic function associated with each role. Then, use these syntactic functions to determine the word forms associated with each role. Lastly, use these word forms to determine the lexical concepts associated with the roles.

Step 8 Bind roles and concepts. Calculate the binding between the lexical concepts and roles and store this binding in the STM module associating the conceptual and schema layers.

We can think of comprehension as proceeding in two passes: a *forward pass* involving steps one to six and a *backward pass* involving steps seven and eight. During the forward pass input signal is broken down into three kinds of representations (syntactic, conceptual and schematic) and each of these representations is stored in the respective memory module. Also, during this pass, the syntactic pattern is associated with the conceptual and schema patterns and stored in the short term memory links. Once this has been done, the backward pass begins. The schema representation is broken down to obtain the roles and the system moves backwards through the layers, using the schema to obtain the syntactic construction, using the syntactic construction to obtain the syntactic elements, using the syntactic elements to obtain the associated word forms and using the word forms to finally obtain the lexical concepts. These lexical concepts can then be bound to the roles to obtain the canonical representation and end the process of comprehension. Figure 5.3.2 shows this forward and backward flow of information. It also shows that, in general, the process of comprehension can be seen as moving from a syntactic configuration to a canonical representation.

Let us describe each of these processes more precisely and formally. The input to the system consists of the binding between a list of word forms $\mathbf{w}_i$ and their associated syntactic elements $\mathbf{s}_i$. Using the tensor-product scheme, we can write

$$\mathcal{L} = \sum_i \mathbf{w}_i \otimes \mathbf{s}_i$$

where $\mathcal{L}$ is the (linguistic) input. The system then uses the syntactic elements to to obtain the representation of the entire syntactic construction. This step requires the system to know the relation of each syntactic element to the whole syntactic construction. We saw in Section 5.2.3 that this meronomic knowledge resides inside the grammar, which we assume is available to us through a lookup table. The grammar contributes a list of relations $\mathbf{r}_i$ which the system binds to each of the syntactic elements to obtain the representation of the whole syntactic construction:

$$\mathbf{S} = \sum_{\mathbf{i}} \mathbf{s}_i \otimes \mathbf{r}_i$$

where $\mathbf{S}$ is the representation of the whole syntactic construction. The system also uses each of the word forms to obtain the concepts $\mathbf{c}_i$. Finally, the system uses the verb to obtain the schema $\mathbf{X}$. Again, we use a lookup table to perform this mapping. But ideally this process will be implemented by a feedforward network that can remember the association between the word form of a verb and the pattern of a schema.

The structures $\mathbf{S}$, $\mathbf{X}$ and $\mathbf{c}_i$ are stored, through Hebbian learning (Equation 5.3.1), in the long term syntactic, schema and conceptual memories respectively. The system also calculates the binding

$$\mathbf{B} = \mathbf{S} \otimes \mathbf{X}$$

and stores it in the short term memory linking syntactic and schema layers. This finishes the forward pass.

During the backward pass, the system first needs to obtain the roles in the schema $\mathbf{X}$. We saw above that we represent the schema as the binding of different roles $\mathbf{arg}_i$ and their positions $\mathbf{pos}_i$. Thus we can obtain each role from the schema $\mathbf{X}$ through the process of unbinding:

$$\mathbf{arg}_i = \mathbf{X} \otimes \mathbf{pos}_i^{-1}$$

where the unbinding operation is shown using the notation of the tensor product between a memory and the inverse of the given vector. As we saw in section 5.2.1.1 (page

201), this unbinding operation can be implemented through the inner product between the memory and the input. Next, we need to find the lexical concept that should be associated with this role and we move backwards through the system to calculate this lexical concept:

$$\begin{aligned} \mathbf{S}' &= \mathbf{B} \otimes \mathbf{X}^{-1} \\ \mathbf{s}'_i &= \mathbf{S}' \otimes \mathbf{r}_i^{-1} \\ \mathbf{w}'_i &= \mathcal{L} \otimes \mathbf{s}_i'^{-1} \end{aligned} \tag{5.3.4}$$

where $\mathbf{S}'$, $\mathbf{s}'_i$ and $\mathbf{w}'_i$ denotes the syntactic construction, syntactic elements and word forms respectively, retrieved through unbinding. These word forms can then be used to obtain the corresponding lexical concept $\mathbf{c}'_i$, for each role $\mathbf{arg}_i$. The system binds each role with the retrieved concept giving us the required canonical representation and thus bringing the comprehension process to an end:

$$\mathcal{CR} = \mathbf{arg}_i \otimes \mathbf{c}'_i$$

§ 5.3.3.2. **Production.—** Production is the reverse mapping, from a canonical representation to a set of words and their syntactic functions – i.e. we are given a binding between roles and concepts and we want to obtain a set of words tagged with their syntactic function. To make things simpler, we assume that the system is given a schema and a list of concepts, which saves the step for unbinding the roles and concepts. The system follows the following steps during production:

Step 1 Simulate the schema and conceptual memories and learn these patterns.

Step 2 Retrieve the short term memory that links syntactic and schema layers.

Step 3 Unbind a syntactic construction. Determine a cue for retrieving syntactic memory by unbinding a syntactic pattern from the short term memory and the schema pattern.

Step 4 Recall syntax. Use the cue to retrieve a syntactic construction from the long term syntactic memory.

Step 5 Bind syntax and schema. Calculate the binding between the retrieved syntactic construction and the given schema pattern. Update the short term memory linking syntactic and schema modules with this binding.

Step 6 Determine syntactic elements that can express the roles. Again, using the schema patterns and the (updated) binding between schema and syntactic pattern, determine the syntactic construction. Unbind different syntactic elements from the syntactic construction and use this to construct a list of syntactic functions.

Step 7 Determine words. Use the input list of lexical concepts to determine a list of word forms.

Step 8 Pair the list of syntactic functions (from step 6) and word forms (from step 7) to obtain the output sequence of words and their syntactic functions.

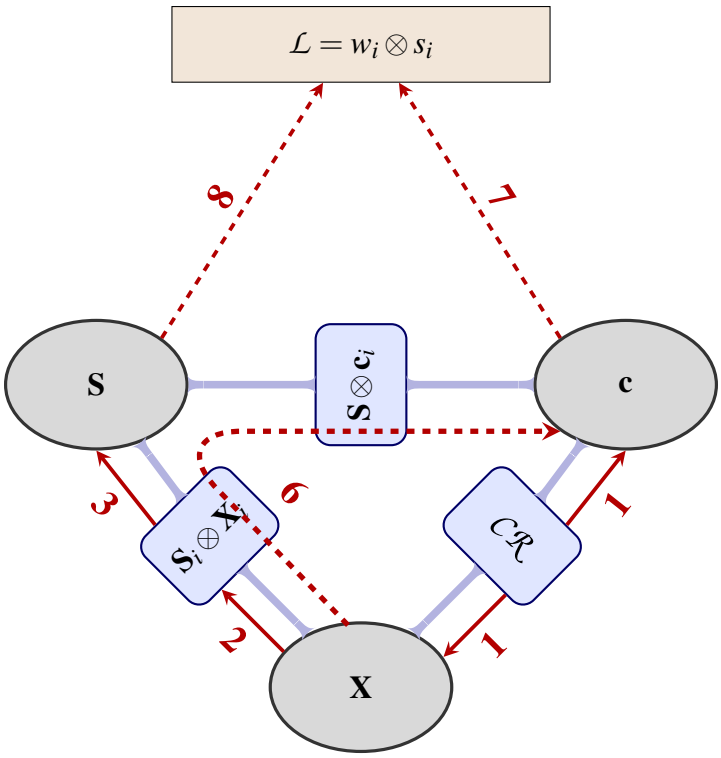


Figure 5.3.3: [Production] This figure shows the flow of information from a canonical representation to a linguistic signal. *Pass one* is shown using bold lines and the *Pass two* is shown using dashed lines.

Just like the comprehension procedure, production operates in two passes. However, unlike comprehension, both these passes are in the forward direction (from canonical representation to linguistic signal) and so we call them *pass one* and *pass two*. Pass one includes steps one to five and just like the forward pass of comprehension involves

simulating the three kinds of memory and obtaining a syntactic construction that corresponds to the given canonical representation. Pass two includes steps six, seven and eight. This pass, not unlike the backward pass during comprehension, uses the patterns across different memory modules to bind word forms to syntactic elements.

Using the same notation as the one presented during the discussion of comprehension, we can see production as the process of retrieving a list of words $\mathbf{w}_i$ and a related set of syntactic elements $\mathbf{s}_i$. This process starts with determining a list of word forms $\mathbf{w}_i$ for the given list of concepts $\mathbf{c}_i$. Next the system determines a syntactic construction $\mathbf{S}$, linked to the given schema via the unbinding operation

$$\mathbf{S} = \mathbf{B} \otimes \mathbf{X}^{-1}$$

This syntactic construction is used as a cue for syntactic memory. The result of the retrieval is stored in the pattern $\mathbf{S}'$ and used to calculate a new binding between syntactic and schema layers, $\mathbf{B}'$. The reason for updating the binding in the short term memory is that the result of the retrieval from syntactic memory might not be the one that is predicted by the tensor binding. Our framework assumes that the syntactic memory is retrieved based on a number of different factors and that the input from the association between syntax and schema memories is only one of them. We can simulate these different factors by assuming that retrieval from the syntactic memory is noisy and we explicitly introduce random noise in the cue for syntactic retrieval. Since the syntactic construction retrieved might not match the results of the unbinding, it becomes necessary to update the binding between schema and (a possibly new) syntactic construction. Later, when the system tries to determine the syntactic elements for the given list of roles, this updated binding will be used. This completes pass one.

The second pass uses the result of the memory simulations made during pass one to calculate the sequence of information structures required to obtain the matched list of words and syntactic elements:

$$\begin{aligned} \mathbf{S}'' &= \mathbf{B}' \otimes \mathbf{X}^{-1} \\ \mathbf{s}_i &= \mathbf{S}'' \otimes \mathbf{r}_i^{-1} \\ \mathcal{L} &= \mathbf{s}_i \otimes \mathbf{w}_i \end{aligned} \tag{5.3.5}$$

where $\mathbf{r}_i$ is the list of relations in the syntactic construction obtained from the grammar and $\mathcal{L}$ is the (linguistic) output this time around. Figure 5.3.3 shows the flow of information during production, labelled with the different information structures generated along the path.

§ 5.3.3.3. **Overlap between Comprehension and Production.—** Having formally characterised the processes of comprehension and production, we can now look at how comprehension and production overlap. We would like to see what micro-processes and representations are common between the two procedures and how flow of information during comprehension can lead to priming during production.

Our first observation is that both comprehension and production are characterised by the flow of information in the same memory modules. Rather than characterise comprehension and production as mutually exclusive procedures that rely on the mental lexicon, we describe both these processes as a set of transformations between three kinds of representations – syntactic, conceptual and schematic. This means that learning in any of these memories during one of these procedures will affect the corresponding memory retrieval during the other. Specifically, step five during comprehension requires the syntactic memory to learn an input syntactic construction. Thus an episode of comprehension will change the configuration (the weight matrix) of the syntactic memory module. During a subsequent episode of production, at step four, the system retrieves a syntactic memory corresponding to a cue. This retrieval depends on the internal configuration of syntactic memory and therefore starts to depend on the syntax of the construction during the comprehension episode, making comprehension and production entangled.

Besides this overlap in the memory modules, the micro-processes of comprehension and production also overlap. One can see this by comparing equations 5.3.4 and 5.3.5, which describe the backward pass of comprehension and pass two of production, respectively. The first two steps in each of these equations are identical: $\mathbf{S} = \mathbf{B} \otimes \mathbf{X}^{-1}$; $\mathbf{s}_i = \mathbf{S} \otimes \mathbf{r}_i^{-1}$. These steps describe the process of transforming a schema pattern into a set of syntactic elements. The production process performs these transformations because it needs to express a set of concepts in a syntactic configuration. These steps give the system the syntactic elements to associate with each concept. The comprehension process performs these transformation because it needs to find a match between the roles of the schema and the concepts expressed in the input utterance. These steps give the system the syntactic configuration of the roles in the utterance, using which the roles can be mapped to the correct concepts in the utterance.

Let us consider an example to illustrate this overlap in the micro-processes of comprehension and production. Let us say the speaker wants to express the schema of **GIVING** such that JOHN is the GIVER, MARY is the RECEIVER and BOOK is the OBJECT GIVEN. This is an episode of production and Equation 5.3.5 requires the sys-

tem to determine the syntactic construction corresponding to the schema of **GIVING**. Let us say that because of the way the memory is 'wired up', the result of the unbinding is a double object dative. The second step in the equation is to determine the syntactic elements corresponding to the double object dative. The unbinding procedure will show that these are *<Subject><Verb><Object 1><Object 2>*. The system can now use these syntactic elements to map the GIVER to the *<Subject>*, the RECEIVER to *<Object 1>* and the OBJECT GIVEN to *<Object 2>*.[7]

Similarly consider a comprehension episode where the system has to comprehend the input *John <Subject> gave <Verb> Mary <Object 1> the book <Object 2 >*. Equation 5.3.4 suggests that the system can determine the concepts that should bind to the roles in the **GIVING** schema by first determining the syntactic construction to which this schema is bound. This unbinding should give the double object dative as the schema has just been bound to this syntactic construction during the forward pass. The second step is to determine the syntactic elements in the double object dative through the unbinding procedure. These syntactic elements are *<Subject><Verb><Object 1><Object 2>*. The system can now retrieve the word forms (and eventually the lexical concepts) linked to each of these syntactic elements and use these to bind to the roles of the **GIVING** schema.

We see that both comprehension and production rely on the same processes of first unbinding the syntactic construction from a schema and then unbinding the constituent syntactic elements from a syntactic construction. Both processes start with the schema **GIVING** and retrieve the list of syntactic elements *<Subject><Verb><Object 1><Object 2>*. This means that the backward pass of comprehension can be viewed as the production process of retrieving syntactic elements to express the given schema. The model suggests that comprehension implicitly involves production. More precisely, the information flow during comprehension and production overlaps.

5.4 Simulation and Results

We are now ready to test the behaviour of our extended model. So far in this chapter, we have proposed a scheme for information processing during language comprehension and production. This scheme describes each of these procedures as a set

[7]Had the memory suggested that the system use a prepositional dative instead, the second step would have given the syntactic elements *<Subject><Verb><Object 2><pp><Object 1>*, mapping the RECEIVER to *<Object 2>* and OBJECT GIVEN to *<Object 1>*, reversing the order of MARY and BOOK.

of transformations converting information structures back and forth between syntactic, conceptual and schema modules. In this section, we would like to test whether this scheme of information representation can explain the patterns of structural priming. We divide this section into three subsections, each of which tests a particular property of structural priming. First we test whether the system shows any structural priming and lexical enhancement of this priming. Then we test whether structural priming accumulates over a set of trials. In the end, we test how long structural priming and its lexical enhancement persist. Each of these examinations is matched with an experimental study and we present a comparison of the results of the experiment and the simulation.

§ 5.4.1 Priming and lexical boost

As our first goal, we examine whether the model shows any structural priming and lexical boost. We simulate the model under the experimental set-up of Pickering and Branigan (1998) who found that subjects show structural priming from comprehension to production during a sentence completion task. They also found that structural priming is stronger when the verb is repeated between prime and target.

§ 5.4.1.1. **Experiment Design.—** Unlike a psychological experiment, where human subjects walk into the experiment room, equipped with their knowledge of language and memory of linguistic episodes, a simulation is conducted on a model whose epistemic characteristics are strictly controlled by the modeller. In order to make a direct comparison between the results of the experiment and the simulation, the model must make its relevant knowledge comparable to that of the human subject, before the testing begins. For this reason, the simulation on the model is conducted in two phases. The first phase consists purely of comprehension trials and presents the model with each syntactic construction, schema and concept in the stimuli. These trials give the model a long term memory of each of these items. In the absence of such long term memories, we cannot measure structural priming because the linguistic selection made by the model might be based on the lack of a linguistic alternatives rather than an assumed choice between alternatives. For example, we cannot test a (hypothetical) subject for structural priming between a PO and DO if the subject has never come across a PO before. While psychological experiments assume such linguistic competence by, for example, choosing only native speakers of a particular language, the model avoids this possibility by presenting each possible linguistic structure during the

first phase of the simulation. We call this phase the *acquisition* phase.

The second phase of the experiment is the *testing phase* and presents the model with a sequence of comprehension and production trials. The results of this phase can be directly compared against the experimental findings and therefore, the design of this phase of the simulation must match the design of the experiment. Mirroring Pickering and Branigan (1998), each testing phase contained a list of 32 items each consisting of a comprehension trial followed by a production trial. Each comprehension trial provided the model with a linguistic signal ($\mathcal{L}$) – i.e. a sequence of pairs of words and their syntactic function – which could be in either a PO or a DO construction. The production trial provided the model a canonical representation – i.e. a schema (consisting of a list of roles) and a matching list of concepts – which could again be expressed as either a PO or the DO. Each production trial could follow either a PO prime (comprehension trial) or a DO prime.

In addition, the schema used during the production trial could either be the same as the verb used during the comprehension trial, or be different. Thus there were four different priming conditions: PO-Same, PO-Different, DO-Same and DO-Different. The testing phase chose eight items under each of the four conditions to give the list of 32 items. The order of the priming conditions was randomized and the verbs were also randomly chosen from a list of six verbs[8]. The input to the model came from an experiment file, which presented a sequence of comprehension and production trials that the model simulated. Figure 5.4.1 shows a fragment of such an experiment file used during a testing phase.

The model was simulated for thirty different subjects. Each subject differed from the other in the acquisition phase which randomised the order in which different stimuli were provided and inserted a random delay at the end of the acquisition phase. This random delay means that different subjects had a variable memory for each of the linguistic constructs. Each subject also received a different list of trials during the testing phase. Each list was randomly constructed under the constraints mentioned above.

The model wrote the output of each trial to two output files. Each comprehension trial resulted in a list of matched concepts and roles (i.e. $\mathcal{CR}$) which was written to a comprehension file and each production trial gave a matched list of words and syntactic functions (i.e. $\mathcal{L}$) which was written to a production file. Figure 5.4.2 shows a segment from a production file for a particular subject.

[8] *give, send, hand, show, make* and *sell*

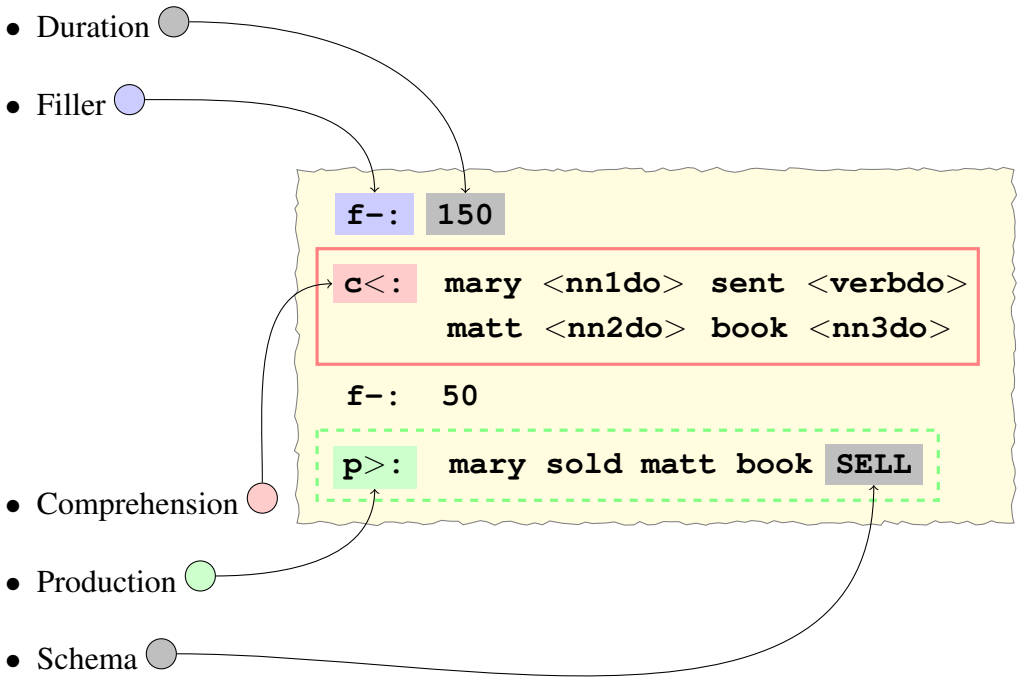


Figure 5.4.1: [Experiment file] Fragment of a file used to test the model. The first character of each line indicates the type of trial (comprehension, production or filler). Thus, the solid box shows a linguistic signal ($\mathcal{L}$) while the dashed box shows a canonical representation ($\mathcal{CR}$). Filler trials show the duration of the filler, comprehension trials show a list of word forms and their associated syntactic function and production trials show a list of concepts and the schema. The tags <nn1do> refers to the subject, <nn2do> refers to the first object and <nn3do> refers to the second object for the DO trial.

The simulations also fixed the free parameters of the model to realistic values. These free parameters included the variables for controlling the rate of learning and forgetting. The parameters for learning in long term memory were the learning rate λ (Equation 5.3.1) and the upper bound of the connection weights W_{max}. These variables control how memory accumulates over a series of trials. On one extreme, if the rate of learning is the same as the upper bound on weights, then the model learns quickly but since the connections achieve their maximum weight after a single trial, there is no cumulative effect of learning. In the other extreme, if λ is much smaller than W_{max}, then the model learns very slowly and the cumulative effect stretches back into the remote past. Thus, what is important is not the absolute value for each of these variables, but

```
JOHN <nn1> SELL <ver> MARY <nn2> DRSS <nn3>
MATT <nn1> HAND <ver> JOHN <nn3> SEAT <nn2>
MARY <nn1> GIVE <ver> MATT <nn3> BOOK <nn2>
MATT <nn1> SHOW <ver> JOHN <nn2> SEAT <nn3>
JOHN <nn1> SELL <ver> MARY <nn3> DRSS <nn2>
```

Figure 5.4.2: [Production file] Fragment from a production file generated by the model. Each line shows a string of words along with their syntactic function. The last line is a PO utterance *John sold the dress to Mary* while the second last line is a DO utterance *Matt showed John the seat*.

how one is related to the other. We chose $W_{max} = 4\lambda$ so that (in the absence of decay) learning accumulated for four successive trials before saturation.

The parameters related to forgetting are the rate of decay in long term memory τ_{ltm} (Equation 5.3.2) and the rate of decay in short term memory τ_{stm} (Equation 5.3.3). In the current set of simulations, we assumed the hypothesis that the lexical enhancement of structural priming is a short-lived phenomenon (Hartsuiker et al., 2008) while structural priming itself persists for a much longer time (Bock & Griffin, 2000). Thus we fixed $\tau_{stm} << \tau_{ltm}$. Since the simulation tested for lexical enhancement from the prime to the target, the duration of filler between prime and target was set to $0.5\tau_{stm}$ and the duration of filler between two consecutive items was set to τ_{stm}. These values were chosen so that there was influence from the short term memory on the selection of a target after a prime, but this influence decreased between two consecutive items.

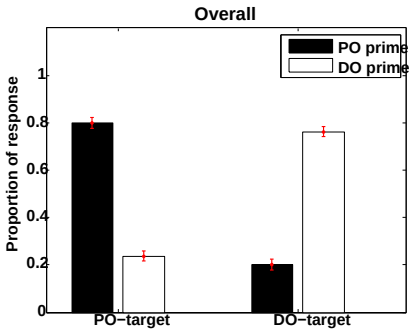
§ 5.4.1.2. **Results and Analysis.—** The results were accumulated in an identical manner to Pickering and Branigan (1998). For each subject, we counted the number of PO productions following a PO prime, the number of PO production following a DO prime, the number of DO productions following a DO prime and the number of DO production following a PO prime. Next, we calculated the *proportion* of PO and DO responses following the PO prime condition and similarly for the DO prime condition. As noted by Pickering and Branigan (1998), this gives us the conditional probability of each kind of response following each kind of prime. We also divided the results based on whether the verb was repeated between the prime and target (Same-verb condition) or prime and target used different verbs (Different-verb condition).

The results of the simulation are presented in Figure 5.4.3. The dependent variable

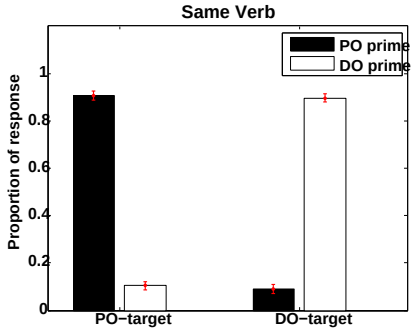
Experiment 1 Results

Verb type	Prime completion	Target completion: PO	Target completion: DO
Same	PO	.47	.22
	DO	.29	.38
Different	PO	.40	.25
	DO	.35	.29

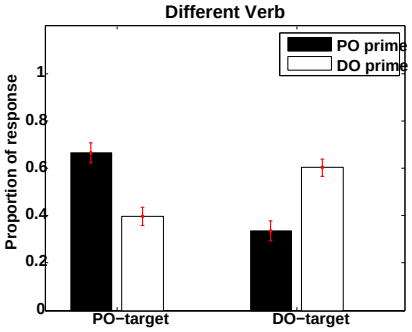
(a)



(b)



(c)



(d)

Figure 5.4.3: [Results] The dependent variable is the proportion of times a particular syntactic construction was produced for both types of primes. (a) Shows the results from the original study (experiment 1) conducted by Pickering and Branigan (1998). (b) Shows overall proportions of PO and DO responses for each kind of prime. (c) Shows the Same-verb condition – i.e. the comprehension and production trials used the same verb. (d) Shows Different-verb condition.

is the proportion of a particular syntactic construction following each kind of prime. The results are divided into the Same-verb and Different-verb conditions and each plot shows the proportion of PO and DO productions following a PO or DO prime.

§ 5.4.1.3. **Discussion.—** The results in Figure 5.4.3 show that the model is more likely to produce a PO after hearing a PO prime (79% of times) as compared to a DO prime (20% of times) – i.e. it shows structural priming. Furthermore, it also shows a larger amount of priming under the Same-verb condition as compared to the Different-verb condition. For example, the difference between the PO and DO targets after hearing a PO prime increases from 37% in the Different-verb condition to 74% in the Same-verb condition. Thus, the model also replicates the results of lexical boost shown by Pickering and Branigan (1998).

In Section 5.3.3 we looked at a scheme for comprehension and production that shows an overlap in the micro-processes and representations of the two procedures. The results of the current simulation show that this scheme for linguistic processing is sufficient to explain the results of structural priming and lexical boost. In particular these results show that an episodic unsupervised learning mechanism can lead to structural priming. It also suggests that lexical boost could be due to the (short term) association between the syntax and schema structures of a construction.

§ 5.4.2 Cumulative structural priming

Having established that our framework for linguistic processing is capable of showing structural priming, we would like to investigate whether the given mechanism of learning can account for the cumulative properties of structural priming. Kaschak and Borreggine (2008) and Hartsuiker and Westenberg (2000) both showed a "long term" effect of priming where priming accumulates over a series of priming trials. To preserve continuity with our analysis of model ③, we tested the extended model for the stimuli developed by Kaschak and Borreggine (2008) and examined the effect of a sequence of priming trials on selection of a syntactic construction.

§ 5.4.2.1. **Experiment Design.—** The above experiment design for testing structural priming and lexical boost (Section 5.4.1.0) is augmented to include three phases in place of two. The first phase, where the model learns the long term memories for all possible stimuli (acquisition phase) remains the same. The testing phase is split into two: a *training phase*, which corresponds to the bias phase in Kaschak and Borreggine (2008) and a *testing phase*, which corresponds to the priming phase in their experiments. The definition of the training and testing phases are same as those employed for testing model ③ in the previous chapter (Figure 4.5.5 on page 151).

The experiments designed by Kaschak and Borreggine (2008) varied the kind of cumulative priming a subject received during a sequence of comprehension trials (the training phase) and examined the effects this training had on the short term priming (i.e. when target immediately follows prime) during a subsequent testing phase. Replicating their design, the model received a list of 10 consecutive comprehension trials (primes), each of which is either a PO construction or a DO construction. This list could contain either an equal number of PO and DO trials, called the *Equal* condition or an unequal number of PO and DO trials, called the *Unequal* condition. When the model received an unequal number of POs and DOs, all constructions were of the same type – i.e. only

POs or only DOs. This list of comprehension trials used verbs that were all drawn from one of two sets. The first set contained the verbs *give* and *sell* and the second set contained the verbs *send* and *hand*.

After presentation of the 10 comprehension trials, the model began its testing phase during which it was presented a list of 6 pairs of comprehension and production trials. All the comprehension trials in a list were either all POs or all DOs. All the verbs for each list were selected from one of the two sets mentioned above and each production trial used the same verb as the comprehension trial.

Thus each production trial made its syntactic decision based on the following conditions: (i) Equal/Unequal primes in training, (ii) PO/DO comprehension trial; (iii) Set 1/Set 2 used during training and (iv) Same/Different sets of verbs in training and testing. Based on these four different variables, a subject could belong to one of sixteen different conditions ($2 \times 2 \times 2 \times 2$). The simulation was run for 64 different subjects, so that there were four subjects under each condition. To run a simulation, we constructed the stimuli files for each of the three phases and the model generated comprehension and production files for the testing phase, which were analysed to obtain the results. A sample file for the testing phase for a particular subject is shown in Figure 5.4.4 (a similar file was also created for the training phase). From this file, we can see that the subject received six groups of comprehension-filler-production sequences. Each comprehension trial was a DO construction and picked the verb from the set {*send, hand*}. This testing phase could have followed a training phase with either an equal number of PO and DO primes (Equal condition), or a training phase with only PO primes (Unequal condition). The training phase could not have been only DO primes because the testing phase always uses the opposite construction for training and testing phases under the Unequal condition.

The last step in the experiment design requires a specification of the free parameters of the model. The duration of of fillers between prime and target and the duration of filler between two consecutive trials during the testing phase were both set to τ_{stm}. Note that we increase the duration of filler between prime and target. This is done to reduce the lexical boost effect to a certain extent as this setup always repeats the verb between comprehension and production trials leading to a large amount of priming. All other parameters are kept at the same values as the first simulation.

§ 5.4.2.2. **Results and Analysis.—** We collected the data in a manner similar to the first simulation. We calculated the proportion of POs and DOs produced in response

```
c<: matt <nn1do> sent <verbdo> john <nn2do> book <nn3do>
f-: 250
p>: matt handed john book HAND
f-: 50
c<: mary <nn1do> handed <verbdo> matt <nn2do> seat <nn3do>
f-: 250
p>: mary sent matt seat SEND
f-: 50
c<: john <nn1do> sent <verbdo> mary <nn2do> dress <nn3do>
f-: 250
p>: john handed mary dress HAND
f-: 50
c<: matt <nn1do> handed <verbdo> john <nn2do> book <nn3do>
f-: 250
p>: matt sent john book SEND
f-: 50
c<: mary <nn1do> sent <verbdo> matt <nn2do> seat <nn3do>
f-: 250
p>: mary handed matt seat HAND
f-: 50
c<: john <nn1do> handed <verbdo> mary <nn2do> dress <nn3do>
f-: 250
p>: john sent mary dress SEND
```

Figure 5.4.4: [Testing file] A sample file used for the testing phase while simulating Kaschak and Borreggine (2008). Each line stands for a trial. Lines beginning with `c<:` show comprehension trials and consist of the linguistic signal $\mathcal{L}$ (list of words and their syntactic function), while lines beginning with `p>:` show production trial and consist of a canonical representation $\mathcal{CR}$ (list of concepts and a schema).

to a PO or DO prime. In this case we also classified these proportions according to the cumulative priming that the model underwent during the priming phase – i.e. whether it received Equal or Unequal number of primes. Lastly, the results were classified by whether the model received the verbs from the same or different sets during training and testing phases – i.e. the Same-verb or Different-verb condition. We would like to point out that the Same/Different-verb conditions in these results differs from that in the previous simulation, where Same/Different-verb condition meant whether the verb is repeated between a prime and a target. Here, the verb is always repeated between a prime and the target. However, the verbs can be drawn from the same or different sets during the training and testing. Since we are interested in whether an overlap in verbs

affects the cumulative nature of priming, the Same/Different conditions here refer to this overlap in verbs during training and testing phases.

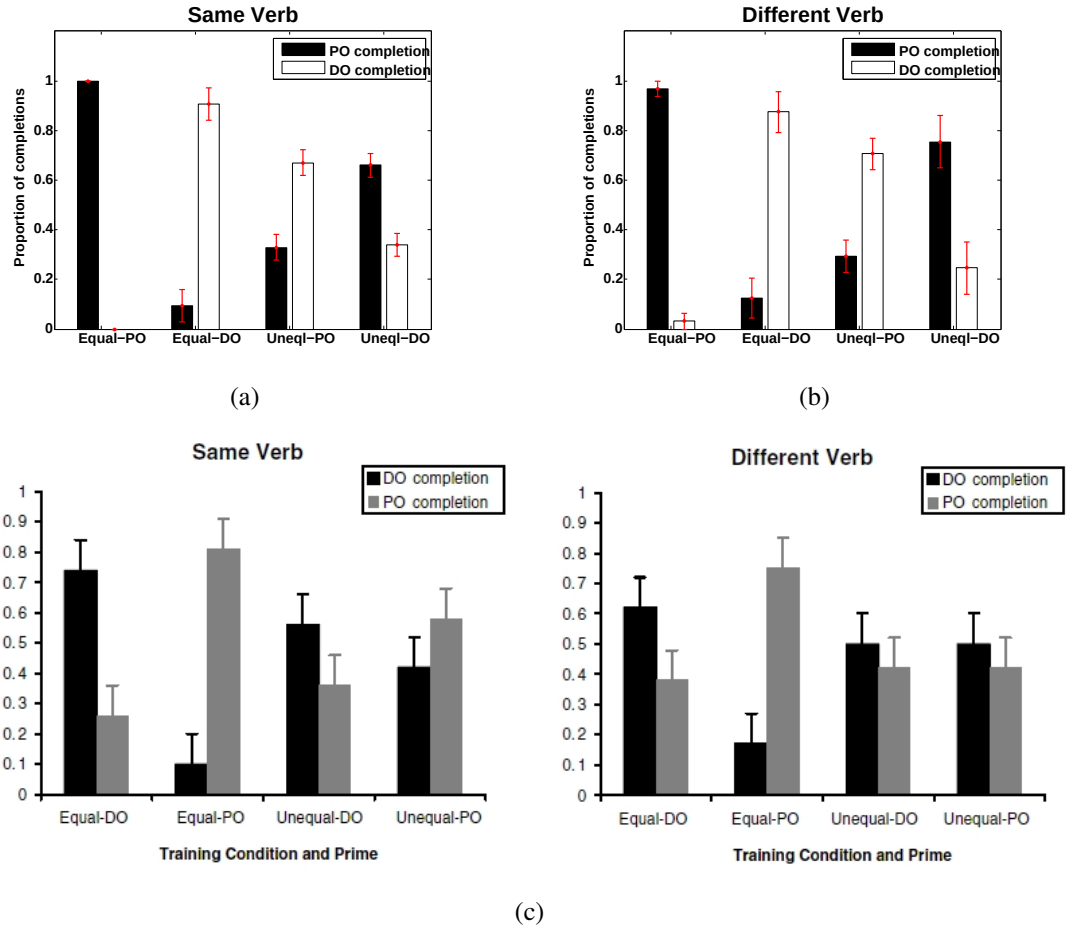


Figure 5.4.5: [Results] The results for Experiment 1 from Kaschak and Borreggine (2008): (a) Simulation results (Same-verb condition); (b) Simulation results (Different-verb condition); (c) Original results from Kaschak and Borreggine (2008). In each case, x axis shows the priming condition and the y axis shows the proportion of each syntactic construction (normalised between 0 and 1).

We present the results of the simulation in Figure 5.4.5(a) and 5.4.5(b). Each plot in the figure shows the (cumulative) priming condition along the x-axis and the proportion of each syntactic construction under each priming condition along the y-axis. Results can be compared both to the experimental study conducted by Kaschak and Borreggine (2008) (Figure 5.4.5(c)) and the results of presenting the same stimuli to Model ③ in the previous chapter (Figure 4.5.8 on page 157).

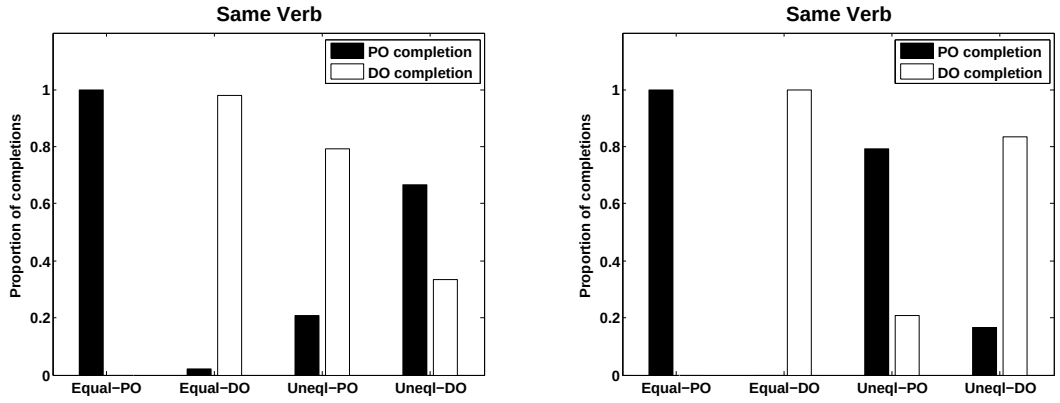
§ 5.4.2.3. **Discussion.—** A comparison of the results of the simulation with the results obtained by Kaschak and Borreggine (2008) reveals a striking similarity. Under the Equal condition, both the experimental study and the simulation show large priming – i.e. when the model is trained on equal number of POs and DOs, it shows a large structural priming during testing. However, this priming is substantially reduced under the Unequal condition, when the model is trained on one kind of syntactic structure, but tested for another structure. Thus priming between comprehension and production during the testing phase shows influence of the sequence of primes during the testing phase. In other words, priming shows a cumulative effect.

The reader may recall that we implemented learning using a modification of the Hebbian learning algorithm with incremental adjustment for each episode and an upper bound on the weights. The results of this simulation show that such an algorithm can explain the cumulative effect of structural priming. Each priming episode activates a long term memory. The more priming episodes that one encounters, the more this memory is consolidated (with an upper bound). During production, the system makes syntactic decisions governed by a systematic analysis and integration of information structures. This analysis and integration of information relies crucially on these long term memories. Since these long term memories have accumulated information from past episodes, the structural decisions made during production show a cumulative effect and come to rely on the sequence of primes.

These results also show similar difference in Equal and Unequal conditions whether or not the verbs are drawn from same or different sets during training and testing (Same/Different verb conditions). Not unlike the simulations on Model ③, these simulations show that the lack of long term lexical enhancement of priming can be accounted using a model where priming and lexical influence on this priming rely on two different kind of memory systems. In the current model, the lexical enhancement of priming comes from the tensor-product memory that binds syntax and schema layers. Since we have proposed that this binding only exists for a short period of time (and hence implemented it using a short term memory mechanism), it is not surprising that the training phase shows no long term effect of overlapping verbs on testing phase.

Furthermore, we can vary the parameters of the model in order to see how internal processes affect the system behaviour. We have concluded that the reduced priming under the Unequal condition (as compared to the Equal condition) is a consequence of the cumulative long term learning in the syntactic layer. We discussed at the beginning of this section that the this long term learning can be controlled using the learning rate

λ (equation 5.3.1 on page 5.3.1). Up till now, we had fixed λ such that $W_{max} = 4\lambda$. This meant that after four consecutive trials of the same type, the weights saturated and the memory did not accumulate any further information. If this is true, then decreasing λ in proportion to W_{max} should increase the cumulative effect of the sequence of (ten) prime trials during the training phase. On the other hand, increasing λ should have the opposite effect – i.e. the priming should start to depend on more recent trials.



(a) $\lambda = \frac{1}{7}W_{max}$ (b) $\lambda = \frac{1}{2}W_{max}$

Figure 5.4.6: [Accumulation] How structural priming changes when we vary the learning rate λ.

Based on these considerations we simulated the model for $\lambda = \frac{1}{7}W_{max}$ and $\lambda = \frac{1}{2}W_{max}$. In the former case, λ was small, therefore we expected greater accumulation (and hence larger decrease in Unequal condition). In the latter case, λ was large and we expected low accumulation (and hence a smaller decrease in Unequal condition). The results for the Same Verb condition are shown in figure 5.4.6. Similar results are obtained for the Different Verb condition – so, in the interest of saving space, we do not show those results here. It can be confirmed from the figure that as we expected, a smaller λ leads to a large effect of the training condition (figure 5.4.6(a)) and hence poor priming for the Unequal condition. In fact, the long term priming from training phase dominates the short term priming during testing phase, so that the model seems to show 'reverse priming' for the Unequal condition. On the other hand, figure 5.4.6(b) shows larger priming under the Unequal condition, when λ was large and the system showed recency.

§ 5.4.3 Persistence of structural priming and lexical boost

Finally, we would like to investigate how long structural priming and lexical boost persist. Bock and Griffin (2000) and Hartsuiker et al. (2008) showed that structural priming seems to persist even when prime and target trials are separated by up to ten filler trials. Hartsuiker et al. (2008) also analysed the comparative persistence of lexical boost and observed that this lexical enhancement of structural priming was prevalent when primes were immediately followed by targets, but disappeared when primes and targets were separated by two filler trials or more.

While we already studied the cumulative effect of priming over a series of prime trials, we also wanted to observe this explicit comparison between priming and lexical boost when the priming is achieved by not a series of prime trials, but by only one trial. Therefore, we simulated the model for the same data as Hartsuiker et al. (2008) and observed how structural priming and lexical boost decay.

§ 5.4.3.1. **Experiment Design.—** Again, the experiment design was similar to the first two simulations where the model was first trained so that it acquired a long term memory and then tested for the designed stimuli. In fact the setup of this simulation remained exactly the same as the first simulation, with two exceptions. Firstly, this simulation changed the 2 (PO/DO) $\times$ 2 (Same/Different) design to a 2 $\times$ 2 $\times$ 3 (Lag 0/2/6) design used by Hartsuiker et al. (2008). Secondly, instead of having eight items under each condition, we reduced the number of items under each condition to two, so that the total length of the list was twenty four items.

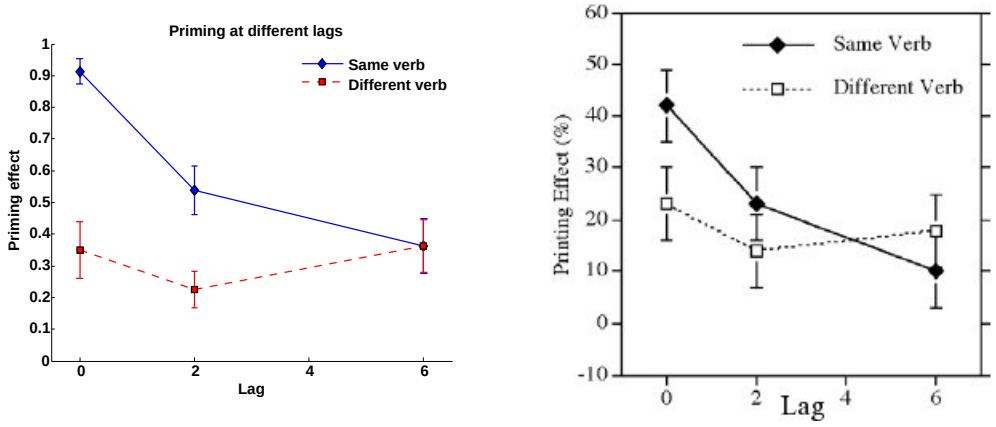
Thus, each list contained pairs of comprehension and production trials where, just like the first simulation, the comprehension trial could be a PO or a DO and the production trial could use the same or a different verb to the comprehension trial. Additionally, the production trial could now follow the comprehension trial by either a zero lag (i.e. no filler trials) or a lag of two or six fillers. Since we encode fillers as the time of decay in the memory, we fixed the duration of one filler to be $1.5\tau_{stm}$. All other parameters remain the same as the previous two simulations.

§ 5.4.3.2. **Results and analysis.—** Like the experiment design, the data analysis for this study was quite similar to the first simulation. Again, we calculated the proportion of PO and DO productions for PO and DO prime. We also classified the results into repeated verb (Same-verb) and non-repeated verb (Different-verb) conditions. Because we want to find out the amount of priming at different lags, we also calculated

this proportion under the three different lag conditions. Lastly, in order to compare the results with Hartsuiker et al. (2008), we calculated the *priming effect* for each construction by subtracting the proportion of productions following the alternative construction prime from the proportion of productions following the same construction prime. For example, the priming effect for PO constructions is:

$$Prop^{\mathsf{PO}}_{\mathsf{PO}} - Prop^{\mathsf{DO}}_{\mathsf{PO}}$$

where $Prop^{\mathsf{PO}}_{\mathsf{PO}}$ is the proportion of PO primes followed by PO targets and $Prop^{\mathsf{DO}}_{\mathsf{PO}}$ is the proportion of DO primes followed by PO targets. The overall priming effect was then calculated by averaging the priming effect for each construction.



(a) (b)

Figure 5.4.7: [Results] Results from (a) simulating the model, and (b) the original study by Hartsuiker et al. (2008) (written session). The x-axis shows the three different lag conditions – 0, 2 and 6 – and the y-axis shows the priming effect. The priming effect for the Same-verb condition is shown using the solid line while that for the Different-verb condition is shown using the dashed line. Note that we normalise the priming effect between 0 and 1, while the original results used percentage.

Figure 5.4.7 shows results for simulating the model for the above data. The x-axis shows the lag and the y-axis shows the priming effect calculated as mentioned above. We observe that there is 94% priming effect at lag 0 under the Same-verb condition, while 59% priming effect for the Different-verb condition. At lag 2, these priming effects change to 57% and 45% respectively. The right hand side of the figure shows the priming effect at lag 6, where it is 50% for the Same-verb condition and 52% for Different-verb condition.

§ 5.4.3.3. **Discussion.—** The key result in Figure 5.4.7 is that we get structural priming for all three lags, while lexical boost diminishes considerably after lag 0. This result is clearly evidenced by the fact that the figure shows a large difference between the Same and different verb conditions at lag 0, but this difference diminishes at lag 2 and lag 6. These results are comparable to the results obtained by Hartsuiker et al. (2008), who also observe that structural priming persists while lexical boost diminishes after lag 0. A sample result from Hartsuiker et al. (2008), for the written session, is shown in Figure 5.4.7(b) for comparison.

These results also mirror the results of the second simulation (above) which also found no long term effect of overlap in verbs between training and testing phases. Our results show that the same internal mechanism of a quick decay in the binding between syntax and schema layers can be used to explain both the quick decay of lexical boost and the lack of cumulative nature of the lexical enhancement of structural priming.

We can confirm that the mechanism responsible for the quick decay in lexical boost is indeed the short term memory connecting syntactic and schema layers by varying the rate at which this memory decays. So far, we have assumed that the time-constant for the rate of decay in short term memory, τ_{stm}, is 100ms. By comparison, we have assumed that the duration of a filler trial is 150ms. This means that the short term memory generates a meaningful cue (step 3 during production) only for about 200-300ms (depending on the noise), after which the syntactic retrieval is completely dependent on the syntactic memory. It is for this reason that a lag of 2 or more filler trials (300 or more ms) does not show significant lexical boost. If this reasoning is correct, then increasing the time-constant of decay in short term memory should mean that the short term memory is able to contribute a meaningful cue for a longer duration of time. Therefore, if we increase τ_{stm} then the schema layer should be able to influence syntactic selection for a longer duration, increasing the longevity of lexical boost.

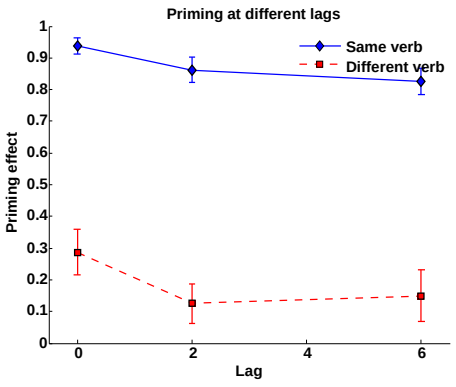


Figure 5.4.8: Results for simulating the model with $\tau_{stm} = 1000ms$.

To confirm this prediction, we increased τ_{stm} to 1000ms and simulated the model with the same stimuli. The results are shown in figure 5.4.8 and can be compared

with our previous results in Figure 5.4.7(a). We can observe that when τ_{stm} is large, structural priming is consistently higher for the Same verb condition as compared to the Different-verb condition. In other words, the model shows lexical boost for all three lags. Thus, the results in figure 5.4.7 are contingent on the rate at which the short term memory decays and these simulations show that the division of labour for implementing structural priming and lexical boost in our model is key to explaining the experimental observations made by Hartsuiker et al. (2008).

Our results do, however, seem to be different from Hartsuiker et al. (2008) in one aspect. In some of their experiments, Hartsuiker et al. (2008) found that structural priming decays steadily as the number of fillers between prime and target trial increase. In contrast, our results show a non-monotonic change in structural priming. In figure 5.4.7(a) the structural priming for the Different-verb condition actually increases slightly, from 45% at lag 2 to 52% at lag 6. The reason for this is the large value of the time-constant for decay in long term memory. We had fixed $\tau_{ltm} >> \tau_{stm}$ which means that over a short amount of lag, the decay in long term memory remains negligible and because the long term memory modules are responsible for priming, we see that structural priming under the Different-verb condition does not change significantly up to lag 6. In other words, τ_{ltm} is too large to let structural priming decrease by a considerable amount for six filler trials. Before we hasten to suggest that the value of τ_{ltm} should be decreased, it should be noted that Bock and Griffin (2000) found no decrease in structural priming for a lag of up to 10 trials and their data is an even better match to our results for the persistence of structural priming. Also, in some of their experiments Hartsuiker et al. (2008) did actually find that structural priming was undiminished for the duration of the experiment. Therefore, it seems difficult to calibrate these internal parameters of the model simply through existing experimental evidence. While the goal of these experimental studies has been to establish that long term persistence of structural priming, the mechanistic account provided in the model demands further investigation into the exact length of this persistence. Such studies would help us calibrate the parameters of the model and explain the cognitive modules shed further light on the physiological modules responsible for structural priming.

5.5 General Discussion

None of the results from the three simulations are surprising. All three results follow from the design of the model. But they do serve as a demonstration of the

model's ability to replicate patterns of structural priming and lexical boost. The model suggests that cognitive processes analyse and integrate information during linguistic processing and transform information from one representation to the other. The overlap of pathways during comprehension and production entangles the two procedures and information processing during one affects the other. The results from the first simulation demonstrate that such an overlap between comprehension and production can lead to priming.

The model's representation scheme proposes that the language faculty can be divided into different modules. Each module specialises in a particular type of representation. In our network, we suggest three modules – syntactic, conceptual and schematic – which transform information between an utterance and its semantics. This modular view of linguistic processing is not a new one and has its basis in both philosophical arguments for the division of the mind along its functional purposes (Fodor, 1983) and physiological findings of the specialisation of information processing in the brain (e.g. place cells in the hippocampus (O'Keefe & Nadel, 1978) and cells in the visual cortex (Hubel & Wiesel, 1962)). We propose that as each module processes linguistic information, it activates, i.e. retrieves, a linguistic structure. This activation of a linguistic structure, in turn, leaves a memory trace. Such memory traces increase the likelihood of the reactivation of the linguistic structure and hence lead to priming. The first simulation demonstrates that unsupervised learning in each of the memory modules can, indeed, lead to structural priming.

§ 5.5.0.4. **Episodic learning and episodic buffer.—** Because learning occurs as a result of a single episode of information processing, we call such learning in each of the modules *episodic learning*. However, this notion of episodic learning needs to be distinguished from *episodic memory* (Tulving, 1995). While we propose that learning in each of the modules is a consequence of a single episode of information processing, we do *not* propose that this learning is the entire memory of the episode. Such an episodic memory would require not only the memory of each of these (syntactic and semantic) components of an utterance, but also how these components are linked to each other. In addition, a memory of the episode requires the mechanism for encoding and retrieval of each of these components and their relationships. Therefore, it is possible that a person lacks the episodic memory of an utterance but still undergoes episodic learning of each of the components as a consequence of processing an utterance. Indeed, Ferreira, Bock, Wilson, and Cohen (2008) observed that subjects

with anterograde amnesia lacked such an episodic memory of the utterance but still showed structural priming. We would like to argue that such priming is a consequence of episodic learning in the syntactic memory module. However, unlike Ferreira et al. (2008), we would not want to claim that this learning is *procedural*. Procedural learning takes place as a consequence of repeated presentation of an input stimulus. Because structural priming is a consequence of a single presentation of a stimulus (prime), we would like to argue that this learning is actually episodic. In other words, we would like to argue that subjects show structural priming not because they adjust their (procedural) rules for grammatical processing, but because they retain an episodic memory trace of the syntax of the utterance. Indeed, during each of the above simulations, the rules of syntactic processing (such as grammatical relations r_i and syntactic mapping ψ) remain unchanged but the model still shows structural priming due to the episodic learning of the activated syntactic structure.

While this modular architecture splits linguistic information into a set of specialised representations, our model also suggests that an integration mechanism binds this disparate information into a whole. This integration mechanism bears a close similarity to the *episodic buffer* of working memory proposed by Baddeley (2000), who suggested that such an episodic buffer temporarily binds information held in long term memory modules into a unitary episodic representation. We propose that, in a manner similar to the episodic buffer, short term memory modules bind (pair-wise) the long term memories activated during linguistic processing. Our model uses a tensor-product representation for storing these bindings, and we suggest that such a binding mechanism might be physically implemented through an activation based memory. Similarly, Baddeley (2000) has proposed that the episodic buffer might be biologically implemented by the frontal areas of the brain, which have been shown to display an activation based memory (Funahashi, Bruce, & Goldman-Rakic, 1989).

The results from our first simulation show that such a binding mechanism is capable of producing a lexical enhancement of structural priming. One can think of this lexical enhancement as being due to a 'flow of activation' from the schema module to the syntax module. But more precisely, our model implements a production procedure which involves obtaining the cue for syntactic recall by unbinding a syntactic memory from the binding between syntax and schema layers. As this cue depends on the activation pattern over the schema layer, the syntactic choice comes to depend upon the schema representation as well. And because the verb in an utterance governs the schema, a repetition of the verb leads to the repetition of the schema which, in turn,

increases the likelihood of the repetition of syntactic choice. This is the lexical boost effect.

We must, however, emphasise the distinction between comparing the episodic buffer and long term episodic memory. The reader might conclude that since our model implements (something like) an episodic buffer, it also stores the episodic memory for each utterance. If this were true, then structural memory in the model becomes a part of this episodic memory and a loss of episodic memory will lead to a loss of the structural memory and structural priming. This conclusion contradicts the findings of Ferreira et al. (2008), who observed that amnesic patients with poor episodic memories, still showed normal structural priming. However, the above reasoning is flawed as it is based on the assumption that the existence of an episodic buffer (that binds long term memories) is equivalent to a long term episodic memory. We emphasise that just like the episodic buffer of Baddeley (2000) stores the bindings only *temporarily*, our tensor-based binding mechanism also contains only temporary (i.e. short term) storage for associations of long term memories. Evidence for such a temporary storage of episodic information comes from studies done with amnesic patients, such as B. A. Wilson and Baddeley (1988), who showed that while all severely amnesic patients show poor long term recall of prose, in some cases immediate recall of prose can be virtually normal. Thus there is a dissociation between a temporary storage of the linguistic episode and a long term storage of such episodes. How this temporary episodic binding gets stored as a long term episodic memory is an open question that we do not address in our model. Due to this dissociation, it is possible that subjects with anterograde amnesia have a problem moving such a short term episodic memory into a long term storage. So while it is possible that these subjects have a damage to their mechanism of storing and retrieving associations between long term memories, they are perfectly capable of binding such long term memories at least temporarily during linguistic processing. Such subjects, like the ones studied by Ferreira et al. (2008), will show poor long term recall of episodes but still be very much capable of generating short term associations between long term memories.

The temporary nature of the episodic buffer, or equivalently the short term nature of the bindings, also casts light on why lexical boost exists only for a short period, while structural priming lasts over a longer period of time. Because lexical boost depends on the syntax-schema binding in short term memory, it lasts only till the activation based binding between these modules exists. On the other hand, the activation of a particular syntax is recorded in the long term syntactic memory and hence lasts for a much

longer period. We implement this difference in the relative durations of the two kinds of memories by fixing the decay time-constants τ_{stm} and τ_{ltm} in such a manner that $\tau_{stm} << \tau_{ltm}$. As a result of these relative rates of decay in the two kinds of memory, the third simulation replicates the results of Hartsuiker et al. (2008) and the second simulation replicates the results of Kaschak and Borreggine (2008), both of which show structural priming to have a long term effect which the lexical enhancement of this priming to be short-lived.

§ 5.5.0.5. **Coarse coding and semantic overlap.—** In all of our simulations, we have assumed a distributed representation for vectors representing syntax, schemata and lexical concepts. We saw during the discussion of the formal representation of the model that these vectors are generated by a function ψ which uniquely maps each symbolic structure to a connectionist representation. In our simulations we have assumed that this mapping is available through a lookup table but in reality this need not be the case. In a cognitive system, each mapping needs to be generated by a combination of learnt procedures and episodes. In a more sophisticated computational system, this mapping can be generated by another neural network or a system of neural networks.

These more sophisticated systems could generate input vectors based on a set of features for each vector. Each lexical concept, for example, could be represented by a combination of a set of features. Different lexical concepts could have mutually exclusive features, or these features could overlap. A coding mechanism that uses an overlapping set of features is called *coarse coding*. This coarse coding mechanism is particularly interesting to us for conceptual representations.

In the current study, we have investigated structural priming and lexical boost for a set of concepts that we assume do not have any overlap. We have encoded this assumption in our simulations by assuming that the vectors representing each concept is linearly independent of vectors representing other concepts. This linear independence makes the processes of retrieval and unbinding simple, but it also prevents us from investigating how such an overlap in semantic space affects structural priming and lexical boost. By making use of distributed representations, our model certainly provides us the framework of testing such an overlap. The long term memory mechanism implemented by the network of excitatory and inhibitory connections allows for a partial overlap in patterns as does the tensor-product binding mechanism. Both these mechanisms reduce performance with interference and this interference increases the complexity of operating the system. Due to this increased complexity we have not

investigated overlapping representations during the course of this study, but the model provides the means for performing such an investigation.

While we mention that an investigation of semantic overlap will make for an interesting future study, we hasten to add that by semantic overlap we mean an overlap in lexical concepts but do not mean an overlap in thematic roles between different verbs. The reason for studying one kind of overlap and not the other is based in the semantic representation chosen by our model. While conceptual overlap can be studied by investigating the overlap in the vectors stored in the conceptual layer, thematic overlap has no straightforward representation in our model since thematic relations are represented independently for each frame in the schema layer.

In our representational scheme, each verb maps uniquely onto a frame. For example, the verbs *give* and *sell* map onto two different frames, say, **GIVE** and **SELL**. Each frame contains a list of roles and a set of relationships between roles. Crucially, the relationships between the roles of **GIVE** and **SELL** do not have an independent semantic representation, but are separately represented within each frame. Thus, the roles GIVER and RECEIVER are related to each other, but this relationship is internal to the **GIVE** schema and is not the same as the relation between SELLER and BUYER in the **SELL** schema. In other words, thematic relationships have no independent existence, but are part of each frame.

This independence of thematic relationships for each schema means that our model predicts that thematic overlap between primes will have no effect on structural priming – a result that has been found by Bock and Loebell (1990). We saw in Chapter 2 (page 25) that this study compares the amount of structural priming for sentences such as *The wealthy widow gave her Mercedes to the church* and *The wealthy widow drove her Mercedes to the church* on sentences that could be produced as, for example, *IBM promised a bigger computer to the Sears store* or *IBM promised the Sears store a bigger computer*. Bock and Loebell (1990) observed that both primes have the same form, but different thematic relationships and when they found that both sentences show an equal amount of priming of the dative sentence (e.g. the *sold* sentence above), they argued that people frame their sentences independent of the properties of its meaning (which includes its thematic relationships).

Our model uses independent thematic relationships for any two sentences with different verbs. Thus sentence such as *The wealthy widow gave her old Mercedes to the church* has no schema overlap with a sentence such as *The wealthy widow sold*

her old Mercedes to the church.[9] Thus the model does not receive a 'boost' from the schema layer either for the locative prime or for the dative prime and hence both types of primes show an equal amount of priming.

Although our model predicts the same result as Bock and Loebell (1990), it gives a different explanation for the independence of structural priming and thematic overlap. While Bock and Loebell (1990) argued that the lack of additional structural priming in case of thematic overlap shows that structural decisions are made without a semantic influence, we argue that syntactic decisions actually are based on semantic properties, but that thematic overlap does not necessarily bring these semantic properties any closer and hence has no effect on structural priming.

5.6 Conclusions

In this chapter we have presented a new framework for language comprehension and production with a view of explaining the patterns of structural priming. This framework is an extension of the models presented in the previous chapter and improves both the representational scheme and the algorithms presented in the last chapter. It provides a detailed mechanistic account of both comprehension and production and how the two procedures overlap. It also implements a fuller account of how an input signal might be converted into a mental representation for that signal during comprehension and how this mental representation could be transformed into a speech signal during production. We simulated this framework for three observed properties of structural priming and found that it successfully replicates the results of each of these studies. While these results are encouraging, we do emphasise that this framework only looks at comprehension and production from the perspective of structural priming and does not try to encode the complete set of processes entailed in each of these procedures. It also looks at an adult speaker – one who has already acquired the linguistic prowess to comprehend and produce utterances – and does not give a developmental account of how these abilities are acquired. In spite of these limitations, the model provides a fertile ground for investigating the causes and properties of structural priming in a mechanistic and formal manner.

[9] Or a more general statement would be that the sentence *The wealthy widow gave her old Mercedes to the church* has as much in common in schema-space with *The wealthy widow sold her old Mercedes to the church* as it does with *The wealthy widow drove her old Mercedes to the church*.

6 Discussion and Conclusions

The aim of this chapter is to review our key findings and to look at them in the broader context of human memory and cognition. The research goal of this thesis has been to investigate the properties of structural priming and our research method has been to construct computational models that can replicate experimental observations. In the previous two chapters we have constructed a series of models. Each of these models extends our knowledge of the cognitive mechanisms responsible for structural priming. In this chapter, we will move away from the specific details and look at the general cognitive principles that underlie the design of these models and the consequences of these principles. We will begin with the general principles of organisation and justify the adoption of the particular architecture of our models. Here we will connect the architecture of our models with the general organisation of information by the human brain. Then we will move to the choice of learning mechanism in each model and the consequences of adopting these learning mechanisms. Again, we will generalise the discussion and see what these learning mechanisms tell us about the place of structural priming in human memory. In the third section, we will give our reasons for choosing dynamical systems theory for studying structural priming. We will briefly review other studies where this theory has been used and see how choosing this theory helps us bring the psychology of structural priming closer to its neuroscience. Finally, we will also list the ways in which we can take our investigations forward and better understand the cognitive underpinnings of not only structural priming, but also language production.

6.1 Specialization and Integration

We begin by explaining the reasons for the functional organisation of various computational models presented in previous chapters. This functional organisation is closely

connected to how the computational models choose to represent information. In this thesis, we have looked at language production (and comprehension) as a flow of information through the cognitive system. The extended model described in Chapter 5 made this view explicit by modelling production and comprehension as a transformation of information (between a linguistic signal $\mathcal{L}$ and a canonical representation $\mathcal{CR}$) as it passes through the language system. This view is also implicit in the three models considered in Chapter 4, which concentrated on the flow of information between the lexical and syntactic representations. When we adopt the view that production and comprehension of speech is equivalent to transforming information, we move our domain of enquiry from the processes of production and comprehension to the mechanisms of information transformation. Instead of asking what the nature of production and comprehension are, we can ask what is the nature of this transformation.

This latter question, about how information is transformed as it passes through the system, can be answered in several ways. It is possible that information is transformed in one step, from an auditory signal to a mental state, however, it could also be transformed in several discrete steps. Furthermore, it is possible that one module is responsible for transforming this information, or it could be that a set of modules transform information independently. The arguments of this thesis rest on the assumption that a number of functionally specialized, independent modules are responsible for *analysing* information and that information analysed in this way is then *combined* using another set of modules. Each of our models makes this assumption. And each model then also assumes that the two types of modules (ones that perform analysis and the ones that perform combination) rely on different kinds of memory. It is in this way that these models are able to explain why structural priming and lexical boost show different temporal properties. Because these assumptions are central to our arguments, it is worth taking a closer look at them – laying out how each of our models makes these assumptions, asking what are the implications of these assumptions for the nature of cognition and explaining how we can justify these assumptions. We take up each of these questions next.

§ 6.1.1 The analysis/combination division in the models

The division between the modules that perform analysis and those that perform combination can be most clearly seen in our final model, described in Chapter 5. In this model, each of the long term memory modules perform functionally specialized pro-

cessing and is responsible for the analysis. Information is analysed (transformed) into three different kinds of representations: syntactic, conceptual and schematic. Each of these modules can be seen to be functionally specialized because the selection of a representation is completely dependent on the previous memories stored in the module. The goal of the module is to transform the input into the memory trace that forms the closest match to the input. The second type of module in the network are the ones that perform tensor-product binding between the conceptual, syntactic and schema modules. By storing the binding, these modules combine this information to form the memory of an episode.

A similar division between two different kinds of modules exists in models ② and ③. The functionally specialized modules in this case are the two winner-take-all layers which implement syntactic and lexical representations. Just like the extended model, the dynamics of each layer is independent of the other and each layer transforms the input signal based on the memory (hysteresis) stored in each layer. The combination between the two functionally specialized layers is again performed by the binding layer. The only exception to this scheme of analysis and combination is our first model, model ①, which has two specialized layers that perform the analysis of the signal, but does not have a layer that performs the combination.

The separation between the two kinds of modules is not just architectural, but also functional. Models ② and ③ assume that the two analysis modules have mutually-inhibitory connections which lead to winner-take-all dynamics. These dynamics are required because these modules have to make a choice between the nodes. The syntax layer has to choose between the PO or DO structures, while the lexical layer has to choose one verb from a set of verbs. These winner-take-all dynamics are in contrast with the dynamics of the binding layer which performs the combination. The nodes in this binding layer are mutually excitatory and as a consequence they are either both active together or both switched off. Unlike the winner-take-all dynamics, there is no competition between the nodes. As a consequence of this contrast in dynamics, the two kinds of modules exhibit different longevities of the memories stored in them. While the winner-take-all layers show exponential decay, the mutually excitatory nodes show catastrophic decay. This difference in longevity of the two types of modules allows us to explain the experimentally observed difference in the longevity of structural priming and lexical boost.

Similarly, the analysis and combination modules in the extended model are also functionally different. The three analysis modules (syntax, concepts and schema) are

implemented using long term memory. This long term memory chooses between different patterns through attractor-network dynamics. The combination modules are implemented using a short term tensor-product memory. This functional division between the two kinds of modules, again, allows us to explain the difference in the longevity of structural priming and lexical boost.

§ 6.1.2 Evidence and reasons for an analysis/combination division

While we have justified the use of the two types of modules based on the longevities of structural priming and lexical boost, we now have to ask if there is actual evidence for the cognitive existence of such a division? First, let us consider the evidence for functional specialization. We discussed in Chapter 2 how evidence for the functional specialization of syntactic knowledge comes from structural priming. Experiments conducted by Bock (1986), Bock and Loebell (1990) and Bock et al. (1992) showed that structural priming exists independent of lexical, thematic or conceptual overlap between prime and target. These results argue for a functionally specialized structural memory, which is separate from memory of other aspects of the prime utterance.

But functional specialization is not restricted to syntactic representations, or indeed to linguistic processing, and is a well-known feature of other cognitive domains. Hubel and Wiesel (1962) found receptive fields in the visual cortex based on single-cell recordings from anesthetized cats. Based on these findings, they argued for a functional architecture of the visual cortex. Since their experiments a number of advances have been made in the study of the visual cortex and the definition and the functions of different functionally specialized areas have been reviewed and refined (see Carandini et al. (2005) for a more recent review). However, the idea of demarcating the visual cortex into these functionally specialized areas (and receptive fields) has endured (Zeki et al., 1991). Similarly, functionally specialized areas have been found within the motor cortex (Talati & Hirsch, 2005) and auditory cortex (Tian, Reser, Durham, Kustov, & Rauschecker, 2001).

Given the existence of this functional specialization in different cognitive domains, we can ask what are the general principles that lead to this method of information representation. One answer comes from the study of how sensory (visual) information is re-represented by the neural system. Our sensory signals arise from natural scenes that are highly structured and consist of a set of well-defined objects, rather than pixels of random activity, or noise. While this sensory data contains all this rich information, the

representation of the sensory data itself is simply a set of colour and brightness values picked by the photoreceptors in the retina. The visual system wants to re-represent this sensory data so that the higher cognitive systems can operate directly on the set of objects, or causes, underlying this sensory data. This problem of extracting a set of underlying causes from sensory data needs to be solved not only by the visual system, but also by the auditory and motor systems.

A number of computational models suggest possible ways in which the statistical information present in sensory signals can be used to learn meaningful re-representations of sensory data (Dayan & Abbott, 2001). All these models accept the sensory data as the input and based on this input, they generate a probability distribution over a set of *latent variables* as the output. These latent variables are interpreted as the underlying causes in a sensory signal. Since these models are able to identify the causes underlying sensory data, they are called *recognition models*. Different recognition models differ in the kind of latent variables that they extract from the input. Models exist that extract statistically uncorrelated latent variables (Principal Component Analysis), latent variables that have Gaussian distribution (mixture-of-Gaussians analysis) and latent variables that are statistically independent (Independent Component Analysis (Bell & Sejnowski, 1995)). Each model relies on a different algorithm and a different property of latent variables, but they all build on the notion that recognition is performed by analysing sensory input into a set of underlying causes or latent variables. From our perspective, these recognition models highlight the reason behind functional specialization in cognition. Friston (2005) hypothesized that sensory information is analysed (or re-represented) by a set of functionally specialized modules because the brain has evolved to extract the causes or latent variables in the input. Furthermore, Friston (2005) pointed out that functional segregation and integration are not mutually exclusive; they are complementary. He discussed evidence that these complementary tasks are performed by *forward* and *backward* connections in the visual cortex. While the forward connections are concerned with the segregation of information (which we have called *analysis*), the backward connections mediate contextual effects and the co-ordination of processing channel (which we have called *combination*).

Our models extend these arguments to the study of language comprehension and production. Just as models of the visual stimuli extract information about objects in a natural scene, our model extracts syntactic and semantic information from a linguistic signal during comprehension. We also study the opposite procedure of production where this segregated information needs to be combined in order to generate a linguis-

tic signal. Thus, the organisation of different modules in our models is not arbitrary, but can be justified based on the general principles of functional specialization and integration pervasive in the human cognitive system.

6.2 Learning mechanism for structural priming

The second major concern of this thesis has been to address the debate surrounding the learning mechanisms underlying structural priming. We have frequently made comments on this debate during previous chapters but here we confront the questions head-on and review how our findings contribute to extending our knowledge on the subject. Here is the obvious question at the heart of the debate: what are the learning mechanisms responsible for structural priming? Note that when we speak about learning mechanisms, we do not simply mean the learning algorithm – this is only one of the questions. Another question is the kind of memory this learning leads to. Here, the debate revolves around the categorisation of structural priming as either *implicit* or *explicit* memory (Ferreira & Bock, 2006). Another related question is the longevity predicted by the learning mechanism: is it *long-term* or *transient*? And yet another question is about the function of this learning: is this learning in service of language acquisition (Chang et al., 2006) or is it in service of aligning mental states of interlocutors (Pickering & Garrod, 2004), or both? Of course, these questions are not completely independent. In this section we will see that evidence which answers one of these questions is frequently misunderstood as evidence which also answers the other questions. When Bock and Griffin (2000) found evidence for structural priming to be long-term, they interpreted these results to mean that priming was also implicit. When Ferreira et al. (2008) found evidence that structural priming was implicit, they argued that it suggested priming was responsible for learning syntax. We would like to argue in this section, that the connections between different aspects of the learning mechanism are not entirely straightforward and our models show a state of affairs where the answer to one question can be independent of the answer to the others.

§ 6.2.1 Implicit, procedural and responsible for acquisition?

Structural priming is a form of memory – it connects two events that are separated in time. A systematic study of human memory has shown that it is not one monolithic structure, but consists of a number of systems (Tulving, 1995; Squire, 2004). Because

structural priming is a form of memory, we can ask, to which of these systems does it belong? This is an important question because different types of memory systems store different kinds of knowledge. If we can determine which memory system structural priming belongs to, we will also be able to determine the kind of knowledge that people extract as a result of structural priming.

§ 6.2.1.1. **Landscape of memory.—** First, let us look at the landscape of memory. The stratification of memory has been under constant debate (see Tulving (2007) for a list of 256 kinds of memory), but there seems to be a consensus on some of the major categories. Tulving (1995) identified five major systems of human memory: procedural, perceptual priming, semantic, primary and episodic (see Table 6.1). *Procedural memory* is a non-representational memory – i.e. it does not store true or false propositions about the external world. Instead, it is dispositional and reflected through performance rather than recollection (Squire, 2004). An example of procedural memory is skill learning such as hand-eye coordination during drawing. In contrast, the other four forms of memory are representational or *declarative* – i.e. it is possible to state the changes due to learning in these systems as falsifiable propositions. *Perceptual priming* (in the Perceptual Representation System) is a form of learning observed in amnesic patients, who lose the ability to recognise previously presented information, but still show facilitation in perception as a result of this information (Schacter, Chiu, & Ochsner, 1993). Both *semantic memory* and *episodic memory* are memories of events and propositions about the world. The distinction between these two kinds of memory comes from the autobiographical nature of episodic memories; while semantic memory is simply a declarative memory of facts about the world, episodic memory shows a capacity to re-experience the event in the context that it originally occurred (Tulving, 1984). Finally, *primary* or *working* memory provides a limited capacity system allowing the temporary storage and manipulation of information necessary for complex tasks such as comprehension, learning and reasoning (Baddeley & Hitch, 1974).

In addition to this categorisation of memory into five different systems, the process of learning these memories can be classified based on the awareness of the acquisition of the memory. The role of awareness in memory has been under investigation for a long time. In a series of experiments, Reber (1967) demonstrated that subjects might be capable of learning without being aware of the process. He asked participants in an experiment to try and memorise a series of letter strings, generated by an artificial grammar. When the experimenter later asked the participants to classify whether

Table 6.1: Major categories of human learning and memory (Tulving, 1995)

System	Other terms	Subsystem	Retrieval
Procedural	Nondeclarative	Motor skills Cognitive skills Simple conditioning Simple associative learning	Implicit
PRS	Priming	Structural description Visual word form Auditory word form	Implicit
Semantic	Generic Factual Knowledge	Spatial Relational	Implicit
Primary	Working Short-term	Visual Auditory	Explicit
Episodic	Personal Autobiographical Event memory		Explicit

new strings were generated by the same grammar, they were able to do this task at a better-than-chance level, even though they lacked the ability to describe the rules for the grammar. Based on these findings, Reber (1967) coined the term *implicit learning* for a form of learning performed by subjects without forming explicit strategies for responding to or recoding the stimuli. Since this original experiment, many studies have been conducted (see Schacter et al. (1993) for a review) which show that subjects can show a facilitation in responding to previously presented stimulus (i.e. can be primed) without being aware of acquiring the memory. Usually, the facilitation of recall is tested through priming while the awareness of acquiring the memory is tested by asking the subjects to recognise the stimulus (Tulving & Schacter, 1990). The difference between the priming and recognition scores then shows a reliance on different learning mechanisms. Furthermore, there is evidence suggesting that amnesic patients

show impaired recognition without any impairment in priming (Hamann & Squire, 1997) signalling a dissociation between the brain areas responsible for implicit and explicit memory. Based on these arguments, different memory systems can be labelled as showing either implicit or explicit learning. Tulving (1995) proposed that the first three systems in Table 6.1 show implicit learning while the last two show explicit learning.

§ 6.2.1.2. **Location of structural priming.—** Having established this landscape of different types of memory, we can now ask, where on this landscape is structural priming located. There are several possible candidate locations. If subjects show structural priming because they are aware of the prime episode, then structural priming is likely to be an explicit memory dependent on the episodic memory system. If subjects are unaware of the memory acquisition during prime episodes, then structural priming is likely to be an implicit memory dependent on perceptual priming or procedural memory. The distinction between perceptual priming and procedural memory lies in the speed of learning. Procedural memory is acquired via gradual learning through repeated exposure, while perceptual priming relies on a single episode of learning (Squire, 2004). If structural priming is a result of learning to gradually extract common elements from a series of inputs, then it should rely on the procedural memory system, and if not, on the perceptual priming system.

As seen in the last chapter, Bock and Griffin (2000) investigated the time-course of priming and found that it does not show any consistent decay when prime and target trials are separated for up to 10 filler trials. Based on these results and previous findings that priming does not depend on an effort to remember the priming sentence (Bock, 1986) and that it does not require explicit attention to the form of priming sentence (Bock et al., 1992), they suggested that structural priming is a procedural memory and relies on implicit learning. One key property of implicit and explicit memories that Bock and Griffin (2000) rely on to make their argument is that implicit memory seems to persist for a longer time than explicit memory. However, in light of our previous discussion, this reasoning seems slightly flawed. First of all, episodic memory seems to use explicit learning and is known to exist for long periods of time (Tulving, 1995). Thus, the longevity of priming does not necessarily imply implicit learning. Secondly, Bock and Griffin (2000) seem to confound implicit learning with procedural memory. Even if structural priming relies on implicit learning, it does not necessarily imply that it relies on the procedural memory system. Indeed Table 6.1 shows three different memory systems that could use implicit learning: procedural, perceptual priming and

semantic. Structural priming could rely on any of the three memory systems.

The extended model presented in the last chapter shows a scenario where structural priming could rely on implicit learning but not on the procedural memory system. We saw that each of the long term memory layers in the model (page 217) relies on a single-shot learning algorithm, where the pattern to be learnt is presented to the syntactic, schema or conceptual layers only once. In contrast, we assumed that the procedural knowledge – such as knowledge of how to order different syntactic functions in a sentence – is stored in a lookup table ψ (section 5.3.2.2). Our model showed structural priming solely by performing learning in the long term memory modules and without making any changes to the procedural knowledge. We say that this learning is implicit because the memory of the complete episode relies on the short term memory modules that bind together the long term memory modules. At the end of the last chapter we discussed that these short term memory modules can be viewed as an *episodic buffer* (Baddeley, 2000) that temporarily stores the memory of the episode. After the pattern of activation is lost from this episodic buffer, the memory of the entire episode cannot be retrieved. Therefore, people asked to recognise a sentence will perform poorly on the task as they no longer possess a representation of the complete utterance. However, they would still show structural priming due to the learning in the long term syntax module, demonstrating an implicit, but *not* procedural memory.

Another study that tries to locate structural priming on the memory landscape is reported by Ferreira et al. (2008) who compared the performance of normal and amnesic subjects on both structural priming and recognition. Much like the studies on implicit learning (Tulving, Hayman, & Macdonald, 1991; Hamann & Squire, 1997), this study found a comparable amount of structural priming for normal and amnesic patients, but the amnesic patients performed worse on the recognition task. Ferreira et al. (2008) argued that the reason behind these results was that the amnesic patients have an impaired declarative memory, but an intact procedural memory. The results can then be explained, they argued, by assuming that structural priming relies on the intact procedural memory while recognition relies on the impaired declarative memory, leading to poor recognition, in amnesic subjects, but a normal amount of structural priming. Based on these results, they proposed that when people comprehend an utterance, they update their abstract relational knowledge in their procedural memory and this learning leads to structural priming. As we have seen at several points in the thesis, this argument is also made by the computational model proposed by Chang et al. (2006) who showed that learning abstract syntactic knowledge through error-based learning

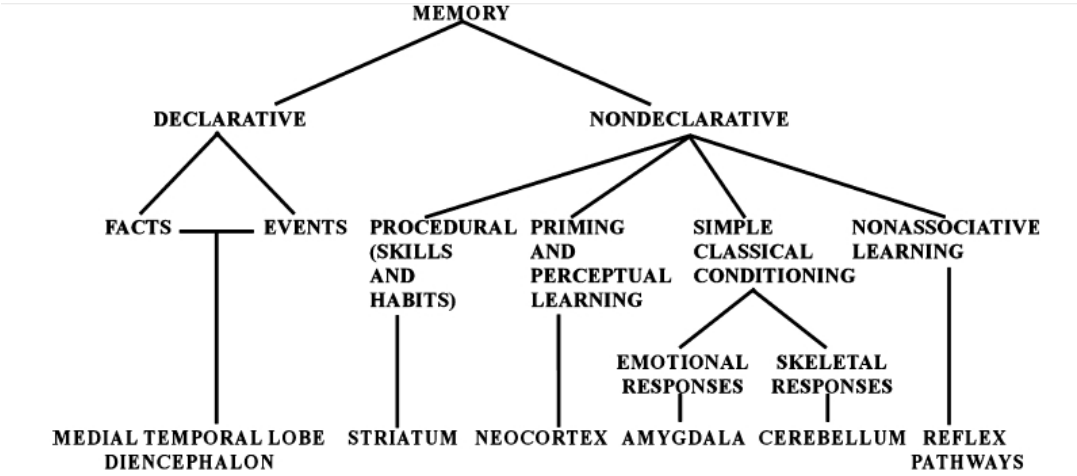


Figure 6.2.1: A taxonomy of mammalian long-term memory systems along with the brain structures thought to be especially important with each form of memory (Squire, 2004).

can lead to structural priming.

But again, this argument relies on a very broad classification of memory – simply procedural versus declarative – a classification which was abandoned in the 1980s in favour of the view that memory consisted of multiple (more than two) memory systems (Tulving, 1995; Squire, 2004). Specifically, Ferreira et al. (2008) assumed that if subjects show an impaired memory for recognition, then they must be relying on procedural memory for performing structural priming. But an impaired memory for recognition could be due to a variety of reasons. For example, damage to the medial temporal lobe is frequently associated with poor recall and recognition (Milner, 1962; Squire, 2004). But as can be seen in Figure 6.2.1 (Squire, 2004), a variety of other undamaged brain areas – striatum, neocortex, amygdala and cerebellum – could be responsible for normal performance of structural priming. Ferreira et al. (2008) assumed that the fact that subjects show structural priming in spite of loss of their recognition memory implies that they rely on procedural memory for structural priming. But this assumption is equivalent to assuming that if subjects show structural priming despite damage to the medial temporal lobe then structural priming must rely on the striatum and not the neocortex, amygdala or cerebellum. Without evidence that the neocortex, amygdala and cerebellum play no part in structural priming, their observations do not merit such a conclusion.

Hence we argue that a loss of recognition memory does not necessarily imply a loss of all forms of declarative memories. The models in this thesis suggest that it is very

much possible that some of the brain areas responsible for declarative memories (such as recognition memory) are also responsible for structural priming. Structural priming in the extended model is due to learning in the long term memory modules. On the other hand, declarative memory for the entire episode is stored in the episodic buffer, or the short term memory modules. This short-term memory is nothing but the binding between different long term memory modules and hence serves as an index, or a pointer into each long term memory. Thus a declarative memory of an episode consists of not only the bindings (pointers) between the long term memories, but also of the long term memories themselves. Thus the long term memory modules participate in both declarative memories and structural priming.

If the memory of these pointers into long term memories is lost, the subject will lose their ability to recognise the episode, without a loss in structural priming. So according to our model, the difference between normal and amnesic subjects is that normal subjects are able to transfer the content of their episodic buffer to a long term episodic memory while amnesic patients cannot perform this transfer. Because normal subjects are able to transfer the contents of their episodic buffer, containing the pointers to the long term memory modules to a long term episodic memory, they are able to use this long term episodic memory at a later point to perform recognition. Amnesic patients, unable to store the contents of the episodic buffer in long term memory are unable to recognise episodes and show poor explicit retrieval.

Thus our model presents a state of affairs where (a) the same modules participate in both declarative memory and structural priming, and (b) these modules are not part of the procedural memory system. We say that these modules are not part of the procedural memory system because unlike procedural memory which performs skill learning through gradual adjustment, these modules perform single-shot learning, storing a component (actually a transformation) of the utterance. Thus, our model presents a state of affairs where the dissociation between structural priming and recognition memory does not imply that structural priming is due to learning in procedural memory, showing that the conclusions made by Ferreira et al. (2008) about the location of structural priming on the memory landscape are too strong.

Support for the state of affairs depicted by our model comes from the study of the hippocampus and its role in the formation of memories. The hippocampus is a part of the medial temporal cortex, damage to which is widely associated with a loss of declarative memory (Squire, 2004). The function of the hippocampus in memory formation is still under investigation, but several studies propose that the hippocampus binds to-

gether different sites in the neocortex (Squire, 1992). Based on this idea, McClelland, McNaughton, and O'Reilly (1995) proposed a computational model for learning in the hippocampus which stores pointers to memories in the cortex and remembers specific episodes by binding together these pointers. By making lesions to this model, McClelland et al. (1995) showed that the model captured patterns of recall shown by amnesic patients. Thus the hippocampus fits the role of a candidate neural system that could store the bindings in the episodic buffer of our model. If this was the case, then damage to the hippocampus would lead to a loss of declarative (recognition) memory in our model, much like it leads to a loss of declarative memory for amnesic patients.

§ 6.2.1.3. **Procedural memory and syntactic acquisition.—** So far our discussion shows that the evidence presented by studies such as Bock and Griffin (2000) and Ferreira et al. (2008) demonstrates that structural priming can be implicit (i.e. without awareness), but it does not demonstrate that it relies on the procedural memory system. Is this an important distinction? We would like to argue that it is. The function of structural priming is contingent on this distinction.

One possible *raison d'être* for structural priming is the learning of the abstract syntactic structure of language. Chang et al. (2006) and Chang, Dell, Bock, and Griffin (2000) showed that a computational model aimed at learning the sequential structure of utterances and how messages map onto this sequential structure will show structural priming. This is a view supported by Bock and Griffin (2000) who argued that the long term persistence of structural priming showed that it could play a role in language learning. This view is articulated in the discussion of their results:

> From a different perspective, the findings make a strong argument for considering an explanation of structural priming in terms of learning rather than transient memory activation mechanisms. The implication of this claim is not simply that a change in performance persists, although it clearly does, but also that the change generalizes to new utterances involving different words. The relevant kind of learning appears to be implicit or procedural...
>
> Chang, Dell, Bock, and Griffin (2000; see also Dell, Chang, & Griffin, 1999) implemented a model that adapts the principles of parallel distributed processing to the circumstances of language production. The model explicitly incorporates a learning mechanism for priming, so that its priming performance depends on the same kinds of weight changes that are involved in its training. In other words, the mechanism of learning is identical to the mechanism of priming. (Bock & Griffin, 2000)

This view emphasises that the same learning mechanism that is used to learn the syn-

tactic structure of language during training (naïve learners) is also the mechanism that is responsible for priming during testing (fluent speakers). Because learning syntax is a form of skill learning as it involves learning a set of implicit rules that are gradually acquired over a series of episodes, Bock and Griffin (2000) assume this learning takes place in the procedural memory. By showing that procedural memory is involved in structural priming, Bock and Griffin (2000) want to show that syntactic learning and structural priming are identical. Thus procedural memory is the glue that binds syntactic learning to structural priming.

However, we have seen that evidence from neither Bock and Griffin (2000), nor Ferreira et al. (2008) merits the conclusion that structural priming is necessarily due to learning in procedural memory. In fact, we have argued that overlapping brain areas might be involved in structural priming and declarative memory and that impaired recall in amnesic patients might be due to impaired access, rather than complete damage, to this declarative memory. This hypothesis is supported by the computational models presented in this thesis. These models demonstrate the possibility that structural priming could be due to learning during testing which is quite different from learning during training. None of the models presented in this thesis tries to learn the procedural rules for generating utterances. The final model assumes that these rules are available from a lookup table that was generated during an (unimplemented) acquisition phase and are unchanged during testing. The only learning performed by the model is in the long term memory modules which, for the syntax module, stores the memory of using a particular syntactic function. Complete loss in this memory would lead to complete loss of priming, but the subject's ability to produce syntactic utterances would remain intact. Thus the mechanism of learning (during acquisition) in this model is *not* identical to the mechanism of priming.

§ 6.2.2 Supervised, error-based and predictive?

Another set of questions about the learning mechanism of structural priming deals with the nature of the learning algorithm. We saw in Chapter 3 that learning could be either supervised or unsupervised. We also saw that one algorithm for supervised learning is error-backpropagation which is used by Chang et al. (2006) to implement their model of language comprehension and production. Finally, we saw that this error required for learning can be generated by making predictions about input data and comparing these predictions with the actual input.

In subsequent chapters we described computational models that have used a different learning algorithm. This learning algorithm is unsupervised – i.e. it does not assume a teacher that can provide the correct response for a given input. For this reason, the model does not generate any error and does not need to make predictions. In addition, this learning algorithm does not try to generate an internal model of the world. Rather, each episode of learning leaves a memory trace in the system which affects future processing. Let us review some of the consequences of adopting such a learning algorithm.

§ 6.2.2.1. **Production-to-production priming.—** One consequence of this scheme is that system is now capable of showing production-to-production priming. The reader may recall that one limitation of the CDB06 model mentioned in Chapter 3 was that it is only able to show comprehension-to-production priming and not production-to-production priming. The reason for this behaviour was that CDB06 requires an external input with which it can compare its prediction to generate an error. This external input is provided by the comprehension trial and not by a production trial. Thus the model performs no learning during a production trial and hence cannot show production-to-production priming.

In contrast, the models presented in this thesis perform learning on both comprehension and production trials. In the extended module, learning takes place in both the long-term and short-term memory modules each time these modules are activated. We assume that each retrieval from long term memory leads to learning – irrespective of whether this retrieval takes place during comprehension or during production. We also assume that the short term memory modules calculate and update their respective bindings during both comprehension and production. Thus, each comprehension and production trial leaves a memory trace in both kinds of modules. Because these memory traces are responsible for priming, the model will show structural priming from comprehension as well as production.

§ 6.2.2.2. **Prediction during comprehension.—** Another consequence of the learning algorithm discussed in our models relates to performing prediction during comprehension. Chang et al. (2006) proposed that subjects actively use their language production system to perform predictions during comprehension. Their model performs incremental comprehension of an utterance, predicting each word and comparing the input word with the prediction. Not only is this prediction mechanism central to learning, it also couples comprehension with production. Thus, their model has a natural

explanation for the overlap in comprehension and production.

In similar vein, Pickering and Garrod (2007) proposed that language comprehension involves using 'emulators' that help the system to predict the linguistic signal based on linguistic and extra-linguistic contexts. The idea of such emulators is taken from the perception literature where it has been proposed that perception of other people's behaviour might involve covert imitation of their behaviour (M. Wilson & Knoblich, 2005; Grush, 2004). Furthermore, Pickering and Garrod (2007) suggested that the comprehension system is likely to use the production system to perform this emulation, rather than implement a separate forward model of its own. They cited neurological evidence which shows an overlap in comprehension and production in support of this hypothesis. Finally, based on this hypothesis, they proposed a conceptual model which uses a production-based emulator to incrementally predict an input signal and uses a Kalman filter to control the amount of influence that emulation has on interpretation.

The extended model presented in the last chapter also showed an overlap between comprehension and production (Section 5.3.3.2 on page 232). However, in contrast to the two accounts above, this model does not require predictions to be made during comprehension. We presented an algorithm for comprehension which involves a forward and a backward pass through the modules. While the forward pass involves an analysis of the input signal into three different kinds of representations, the backward pass involves integrating this information and associating the roles in schemata with syntactic elements. This process of associating roles in a schema with syntactic elements is actually a production process. The production algorithm also operates in two passes and this backward pass of comprehension largely overlaps with pass two during production. In this way the model can be construed as performing emulation during comprehension, even though it does not perform prediction. Models ①, ② and ③ do not specify the processes of comprehension and production overtly, but assume that the same modules (WTA layers) are used for both production and comprehension.

The crucial difference between our proposal and a prediction-based account is that a predictive account hypothesises the use of a production system ahead of hearing speech, while our account supposes that the two systems are used at the same time. In fact, we do not suppose that the two systems are separable and comprehension *is*, in a certain sense, partly production.

§ 6.2.2.3. **Implicit but not error-based.—** A final point about our learning algorithm is that it helps decouple the notions of implicit learning and error-based learning. In Chapter 3 we saw several disadvantages of an error-based account of structural priming. One of those disadvantages was the trade-off between the amount of learning and the amount of structural priming. An error-based account needs to keep the amount of learning low because it wants to gradually extract a model of the syntactic rules of a language from training data. However, as we illustrated, this low rate of learning should lead to a low amount of structural priming and an error-based account has difficulty explaining the amount of structural priming observed in some existing experimental data (see section 3.4, page 70 for details). On the other hand, the gradual extraction of syntactic rules through error-based learning shows how subjects could be learning these syntactic rules without being aware of this process; having extracted the rules, the subjects can discard their memory of the utterance itself and therefore show no awareness of coming across the utterance. Therefore, an error-based account has the advantage of being able to naturally explain why structural priming shows implicit retrieval.

The unsupervised learning algorithms that we have proposed for our models do not face a trade-off between priming and learning rate because these models do not aim to extract the syntactic rules of a language. But in the absence of such a process of rule extraction, how can we explain that structural priming shows implicit retrieval? As we discussed in the previous section, the answer comes from the dissociation between the two kinds of memories in our models. Short term memory is crucial for recognition and recall of an utterance, while long term memory modules can show structural priming on their own. When short term memory is lost, subjects will show failure to recognise the utterance, even though they demonstrate structural priming. In other words, structural priming will show implicit retrieval.

Thus our account shows that error-based learning, or indeed any form of learning that tries to gradually extract a model of the world from stimuli, is not crucial to implicit learning. Such learning is required for extracting procedural knowledge, but as discussed in the previous section, there is no definitive evidence that structural priming is due to learning in the procedural memory system.

§ 6.2.3 Base-level and spreading activation

The idea of a division between a long-term and a transient learning mechanism has led to another recent model proposed by Reitter (2008). This model is based on the ACT-R cognitive architecture developed by J. R. Anderson (1993). Reitter proposed that structural priming has two components, or adaptation effects: a *short-term priming* effect which lasts from one utterance to the next and a *long-term adaptation* effect which persists over a sequence of utterances and for a longer duration. This division between short-term priming and long-term adaptation is akin to the difference in the longevities of lexical boost and structural priming that we have been reviewing in this thesis. Reitter presented evidence from corpus studies that corroborate this division between the two adaptation effects.

In order to establish the cognitive bases of structural priming, Reitter (2008) developed a cognitive model of language production using the ACT-R framework. The basic assumption at the heart of the ACT-R theory is that complex cognition emerges from the interaction between declarative and procedural memory (J. R. Anderson, Budiu, & Reder, 2001). Human knowledge is represented in two separate modules: (i) a *procedural memory* which contains units called production rules governing the transformation of information underlying cognition, and (ii) a *declarative memory* which contains units called chunks that record the information transformed and generated by the production rules. While chunks store factual information (such as "the sum of three and seven is ten"), production rules store procedural knowledge such as mathematical problem solving skills (for example, "if one wants to multiply n1 with n2, then one can add n1 to itself n2 number of times").

ACT-R is particularly useful for understanding the process of knowledge retrieval from memory. Knowledge retrieval is a complex problem because of the amount of knowledge in human memory. ACT-R uses the theory of rational analysis (J. R. Anderson, 1990) to formalise the process of retrieval. According to this theory, the cognitive system identifies the chunks and production rules in human memory that are most likely to be useful in the current context. ACT-R formalises the notion of 'being useful' by calculating the *activation value* of items in memory. This activation value consists of two components: a *base-level* activation which quantifies how useful the item has been in the past and *spreading activation* which quantifies the likelihood of item being useful in the current context. The item with the largest activation value (after accommodating noise) is selected for retrieval.

Reitter (2008) developed a language production model which assigns a syntactic structure to a semantic description using the ACT-R scheme of knowledge retrieval. This model assumes that declarative memory stores not only lexical information (as lexical chunks), but also associated syntactic categories (as syntactic chunks). For example, the lexical chunk for the verb *gave* contains pointers to two kinds of syntactic chunks: a 'ditransitive-to' chunk that accepts a prepositional-phrase complement and a 'ditransitive' chunk that accepts a double-object complement. The syntactic structure of an utterance is governed by which of these two chunks is retrieved during production. Each time a syntactic chunk is retrieved, its base-level activation increases leading to an increased probability of the chunk's selection for future retrieval. Reitter (2008) argued that this increase in base-level activation explains the long-term adaptation effect in structural priming because an increase in base-level activation records the long-term usefulness of the item in memory and, as such, lasts a long period of time.

A third part of the ACT-R architecture (besides declarative and procedural memory) is a set of buffers that store the current state of the system. According to Reitter (2008) these buffers perform a key role in explaining lexical boost. Chunks and production rules retrieved from declarative and procedural memories are temporarily stored in these buffers. Therefore, these buffers can be seen as a simplistic working memory. In order to explain short-term priming and the lexical-boost effect, Reitter (2008) assumed that lexical and syntactic chunks retrieved during one utterance, are retained in these buffers across utterances. Furthermore, he assumed that chunks that are co-present in the buffers will undergo *associative learning*. Due to this associative learning, activation from a lexical chunk will spread to a syntactic chunk during the target trial, if the same lexical item is used during prime and target. Therefore, a syntactic chunk will show an enhanced total activation (sum of base-level and spreading activation) if a lexical item is repeated between prime and target, which is the lexical boost effect. Because associative learning takes place between all lexical and syntactic chunks present in the buffers and not just between the head (the verb) and the syntactic chunk, this explanation of lexical boost predicts that lexical boost will be shown by repetition of any word between utterances and not just repetition of the head. Reitter (2008) confirmed this prediction by performing a corpus-study done on the Switchboard corpus. He found that while repetition of a word strengthens decay, the repetition of a head does not, concluding that any lexical repetition boosts priming rather than specifically head repetition.

§ 6.2.3.1. **Similarities and differences with our models.—** Although the model developed by Reitter (2008) uses a different framework to ours (ACT-R rather than dynamical systems), it comes to a similar conclusion regarding the two learning mechanisms with contrasting longevities underlying structural priming. Both our model and the one based on ACT-R cognitive architecture propose a long-term adaptation mechanism that is responsible for structural priming itself and a short-term priming mechanism that is responsible for lexical boost. Furthermore, these accounts also share the idea of activation spreading from lexical representations to syntactic representation through associative links.

However, there are some important differences in the two accounts. These differences stem from differences in computational principles used for the two accounts and from the difference in their architectures. In an ACT-R model, retrieval from memory depends on activation. Therefore, learning in such a model can be performed by adjusting the value of this activation. Because the activation has two components (base-level and spreading activation), learning relies on adjustment to each of these components. Reitter's model attributes long-term adaptation and lexical boost to these two components and uses this separation to explain the difference in their longevities. In contrast, the dynamical systems outlined in this thesis perform learning through either an adjustment to input sensitivity, or due to hysteresis. Model ③, for example, proposes that cumulative learning relies on the former mechanism (adjustment to input sensitivity) while learning between a prime and trial depends on hysteresis. Simulations performed on the model also show that learning performed amongst competitive (i.e. mutually inhibitory) nodes show a different rate of decay as compared to learning performed in STM (mutually excitatory) nodes. We use this difference in decay to explain the difference in longevity of priming and lexical boost. Thus, even though the ACT-R account and our account attribute the difference in learning to a difference in the underlying learning mechanisms, the two accounts differ significantly in the computational principles that they propose to underlie each phenomenon.

One consequence of this difference in computational principles is each model's prediction about the cognitive mechanism responsible for priming and lexical boost. While our models propose that the syntactic and lexical representations themselves undergo a long-term learning and only the association between these representations (the binding nodes) are held in a short-term memory, Reitter (2008) proposes that the lexical items themselves are held in a short-term memory (buffers) and survive from one utterance to the next. If these lexical items did not survive in the buffers from

one utterance to the next, then Reitter's account would not be able to explain why the longevity of lexical boost is shorter as compared to structural priming. Reitter (2008) argued that it is justifiable to assume that lexical items survive in buffers for a short period of time to achieve coherence between utterances, leading him to predict that coherence would increase lexical boost – i.e. sentences that continue a topic would be more likely to show lexical boost than sentences that do not. In contrast, our model assumes that it is the episode itself (i.e. the binding between long-term representations) that survives for a short period of time and the lexical representations do not need to be held in this short-term memory in order to show lexical boost. Therefore, our account does not rely on coherence for displaying lexical boost.

We also saw above (section 6.2.1) how our account distinguishes implicit learning from procedural memory and explains why the amnesic patients tested by Ferreira et al. (2008) show structural priming comparable with controls but an impaired recognition memory. The ACT-R model proposed by Reitter (2008) can explain these results if it assumes that structural priming is due to learning in the procedural rather than declarative memory. The account can then assume that impaired recognition memory in amnesic patients is due to impaired declarative memory. Since structural priming relies on the unimpaired procedural memory, it will remain undiminished. However, the model proposed by Reitter (2008) does not assume that structural priming relies on learning in procedural memory. Instead, it proposes that structural priming (specifically, its long-term adaptation) is due to the adjustment of base-level activation of syntactic chunks. These chunks reside in declarative memory and a damage to declarative memory would lead to impaired retrieval of these syntactic chunks, which, in turn would lead to a decrease in structural priming, contradicting the experimental results. Moreover, if the ACT-R model was to assume that priming is indeed due to adjustment in base-level activation of production rules (in the procedural memory) rather than adjustment to base-level activation of syntactic chunks (in declarative memory), it would make the same assumption as the model proposed by Chang et al. (2006) and lump implicit learning with procedural memory. As we argued in section 6.2.1, our account allows us to make the distinction between implicit learning and procedural memory. Specifically, we argued that structural priming could be due to implicit learning without relying on learning the rules stored in procedural memory. An ACT-R model is not capable of making such a distinction because of its broad classification of memory into simply procedural and declarative modules.

Finally, the model proposed by Reitter (2008) performs incremental processing

using Combinatorial Categorial Grammar (Steedman, 2000) to generate the constituent structure of utterances. It shares this feature of incrementality with CDB06, which also processes sentences one word at a time. In contrast, our models do not assign a hierarchical structure to the constituents in an utterance, picking the syntactic structure from a lookup table. We will discuss below (in section 6.4) how our model can be extended in the future to replace this lookup table with a module that generates the hierarchical structure of an utterance.

To summarise, a production account based on the ACT-R cognitive architecture shares several features with the models proposed in this thesis. Crucially, it arrives at a similar conclusion about structural priming relying on two different kinds of learning mechanisms. However, we also see that such an account has important distinctions to our models stemming from the different computational principles and architectural choices underlying the two accounts.

6.3 Models and experiments

Dynamical systems theory is a formalism used to study change in the qualitative behaviour of physical systems. In this thesis, we have shown that this theory can also be useful for studying change in a linguistic system, as a result of comprehension or production. The most useful aspect of this theory, with respect to structural priming, is that it allows us to track the behaviour of the system through time, providing a detailed description of how information flows from the prime to the target through time.

But the application of dynamical system theory to the study of biological systems is not unique to our study. This theory has been applied to study, mathematically, how both single neurons and groups of neurons perform cortical information processing. Hodgkin and Huxley (1952) applied dynamical system theory to show how a spike of neural excitation travels through the membrane of a single cell. Hoppensteadt and Izhikevich (1997) described how the theory of dynamical systems can be used to study the different behaviours of neurons ranging from generation of single spikes to oscillatory behaviour where neurons repeatedly generate spikes at fixed time intervals. H. R. Wilson and Cowan (1972) showed how populations of excitatory and inhibitory neurons can show qualitatively different behaviour in response to different class of stimuli. They also showed how particular types of stimuli can lead to oscillations of activity in neuronal populations, providing key insight into how neural tissue might encode different types of stimuli. Amari (1977) showed how neural ‘fields’ with different stability

characteristics could arise within a population of neurons depending on the kinds of connections between the neurons and the type of stimuli given to these neurons. These examples show that dynamical systems theory has a rich tradition in the study of neural behaviour and its dynamics.

§ 6.3.1 Why use dynamical systems theory?

We had one pragmatic reason and one technical reason for using the dynamical systems theory for studying structural priming. The first, pragmatic, reason is evident from the other studies listed above that use dynamical systems theory. These studies describe the behaviour of neurons or populations of neurons. By adopting the formalism of dynamical systems, our study paves the way to making a connection between the psychology of structural priming and its neuroscience. So far, very little is known about the characteristics of our neural system that make it capable of showing structural priming during linguistic processing. If our hypothesis about the computational mechanisms of structural priming is correct, then we can understand something about the neural implementation of this phenomenon by looking at neural systems that are believed to have similar computational mechanism. For example, one key property of the computational models presented in Chapter 4 is winner-take-all dynamics. In order to understand which neural systems lead to structural priming, we can look at the systems that demonstrate this property. Of course this is a very simplistic view and there are all sorts of neural systems that might be able to display this property and extensive research will be required before we can have any clarity on this issue. But, at the very least, our models suggest a possible avenue for investigating the common computational principles behind the relevant psychological and neural processes.

The other reason behind using the theory of dynamical systems is that it provides a form of learning suitable for making the connection between structural priming and trailing activation. The notion of 'trailing activation' is a conceptual and not a formal one. The idea is simple – whenever a part of a system gets activated, some residue of this activity remains in the system after the system has finished processing. This residual or trailing activation interferes with future processing. We saw in Chapter 2 that the spreading activation theory formally defines how activation decays, but that this definition of decay in activation is far too quick to explain structural priming, which seems to survive for at least 10 filler trials (Bock & Griffin, 2000). The alternative to such rapidly decaying structural priming is provided by a process of gradual learning, which

extracts general properties of signals from the environment through repeated exposure. Both these formalisations – rapidly decaying activation and gradual learning – seem to be poor matches for the conceptual notion of trailing activation, which suggests a memory trace that is temporary, yet not transient.

Dynamical systems allow us to capture this idea of change in the behaviour of the system as a result of information processing. Central to the study of the dynamics of a system is the analysis of its stability. When a system is initialised in a state proximal to a point of stability, it tends to approach this stable state. If the flow of information through the system can change where the point of stability lies, then it can change the state of the system to this stable state. We also saw that once a dynamical system is taken beyond a particular point, called the bifurcation point, then the system shows a reluctance to leave its stable state – a phenomenon known as hysteresis. Because of hysteresis, the dynamics of the system come to depend not only on the current state and the input, but also on the history of the system, giving the system a memory. The last link in the chain is to assume that information processing, such as comprehension, pushes the system beyond the bifurcation point, changing its stable states and hence leaving a memory of the flow of information. Such a memory fulfills the criterion of lasting from one episode of information processing to the next without trying to gradually extract the structure (e.g. syntactic rules) of the environment.

As we have shown, this mechanism can lead to structural priming and we can mathematically track the decay in structural priming by specifying how the system adapts or loses its hysteresis with time. Parameters that govern how quickly the system learns and how quickly it adapts allow us to match the pattern of structural priming to the one observed in experimental data. In return, these parameters tell us something about the system. The learning rate tells us how the system changes its sensitivity as a result of information processing. For example, we saw in the last chapter that when the system has a large learning rate, its linguistic choices are governed by more recent information and hence recency becomes a dominant characteristic of memory (page 244). The rate of decay tells us how long the consequences of an episode of information processing last in the system. Even more interesting than the rate of decay is the shape of the decay function. We saw that contrasting shapes of decay function show contrasting longevities for structural priming and its lexical enhancement (section 4.6.2 on page 164). Thus, the theory of dynamical systems provides a framework using which we can formalise the idea of trailing activation and test the computational constraints on the parameters of the formal model that allow it to replicate experimental findings about

structural priming.

§ 6.3.2 Consequences for experimental studies

The consequences of our computational model are not restricted to the formal apparatus for studying structural priming, but also feed back into its investigation through psychological experiments. Through the thesis we have mentioned the predictions of the model for structural priming. Here we review some general properties of structural priming that should be kept in mind when designing future experiments.

§ 6.3.2.1. **Priming could be nonlinear.—** Cumulative priming over a series of trials is implemented by model ③ in Chapter 4 and the extended model in Chapter 5. In both these chapters priming accumulates in a nonlinear fashion. For model ③, we saw that an equal number of each type of prime during training did not mean that the model will be primed equally for both types of primes. The amount of priming for each construction will depend on the sequence in which the constructions are presented. For example, both the sequences PO—PO—DO—DO and PO—DO—PO—DO contain two POs and two DOs but might show different amount of priming because the first sequence presents primes of the same type consecutively, while the second sequence presents the two types of prime alternatively. The amount of priming incurred from a particular trial depends not only on the trial itself, but also the context in which the trial is presented. If a PO trial is presented after another PO trial, then it is likely to incur a larger amount of priming than when it is presented after a DO trial (section 4.6.3).

Note that this prediction is the opposite of the one made by an error-based account. We saw in Chapter 3 that an error-based account predicts that learning is larger when consecutive trials have different structures (Section 3.4.1.4). We also noted in that section that Hartsuiker and Westenberg (2000) presented some evidence from their study that showed priming to be stronger when two primes were of the same type as compared to when the two primes were of different types. This data seems to support the prediction from model ③, which shows larger priming for two consecutive trial are of the same type. In contrast, an error-based account predicts that the error generated by the second trial will be small when it is of the same type as the first trial and the error will be large when the second trial is of a different type to the first trial. Because priming in an error-based model depends on the error, this model predicts a small priming from two consecutive trials of the same type and a large priming from consecutive trials of different types. Therefore, the predictions of an error-based account seem to be

the opposite of the results reported by Hartsuiker and Westenberg (2000). However, we do note that more investigation needs to be carried out on this behaviour because the result reported by Hartsuiker and Westenberg (2000) is an isolated one and their experiment was not designed to measure this behaviour.

The extended model in the last chapter also predicts a nonlinear accumulation of priming. The learning mechanism used in this model is different to the one used by model ③ and therefore it shows nonlinearity for another reason. The reader may recall that we used a learning algorithm which performs cumulative learning with bounded weights (Equation 5.3.1 on page 221). This equation ensures that the weights connecting the nodes in a long term memory module do not grow without an upper bound. This means that priming will grow substantially if a construction is presented consecutively for the first few trials, but after some time it will show saturation. Thus a given scenario in which the model is still 'naïve' (i.e. has undergone little training), a single prime trial might lead to a larger change in priming, as compared to a scenario in which the model has recently undergone training for the construction. Note that this is not the opposite of the prediction made by model ③ – i.e. we are not saying that the sequence PO—DO will lead to a larger priming for DO than DO—DO. Like model ③, the extended model predicts that priming will be larger in the latter case. But, in addition, the extended model predicts that the *increase* in priming for the earlier DO prime might be larger than the increase in priming after the latter DO prime. We say that the increase in priming *might* be larger, rather than will be larger because our model implements a fixed upper bound and whether or not the priming increases by a lesser amount depends on how close the weights are to the upper bound. If the weights are quite close to the upper bound (closer than the learning rate), then the increase in priming will, indeed, be lower for the latter prime. However, if the weights are not yet close to the upper bound, then the increase in priming as a result of the two primes will be equal.

§ 6.3.2.2. **Priming is a multi-headed beast.—** Our account of structural priming can be considered a multi-system account. The extended model presented in the last chapter implements two kinds of modules in the system[1] – the long term memory module, which remembers syntactic elements chosen during a trial, and the short term memory module which remembers the binding between syntactic constructions and schemata. We can think of the long term memory module as being part of a long

[1]The same is true for models ② and ③ from Chapter 4, but the different types of modules are most clearly apparent in the extended model.

term declarative memory system (this could be the perceptual priming system in the categorisation proposed in Table 6.1), while we can think of the short term memory modules as being part of episodic buffer in the working (primary) memory. Besides these two systems, the selection of syntactic structure is also governed by syntactic rules (such as subcategorization frames and ordering of syntactic elements in an utterance) which we have not implemented in our system – or, more precisely, have implemented but as a simple lookup table. As shown by Chang et al. (2006), this kind of memory will also undergo change as a result of the prime trial and therefore contribute to structural priming as well. This change will be a gradual adjustment of the procedural knowledge and is likely to be implemented by a third kind of memory system – procedural memory. Thus, the results of information processing during the prime trial are likely to cause changes to, at least, three different kinds of memory systems.

What does such a multi-system account of priming mean for experimental investigations? Can the experimenter afford to overlook the distinction between the different systems? We argue that a multi-system account has consequences for two aspects of psychological experimentation – its design and the interpretation of results. With regards to experiment design, the experimenter would need to be careful about two aspects. The first are the materials used in the experiment. Different materials used in the experiment might lead to variable contributions from different memory modules. For example, materials that assist the participant's explicit retrieval would lead to contributions from both explicit and implicit memory towards structural priming and consequently lead to larger amounts of priming as compared to experiments that do not assist explicit retrieval. We have suggested that the lexical boost (Pickering & Branigan, 1998) and semantic boost (Cleland & Pickering, 2003) are two examples where explicit memory aids recall. The second aspect of design that may be affected by the multi-system account is the procedure of the experiment, specifically the temporal distance between prime and target. Since the memory in the short term memory modules decays quickly, any procedure that tries to measure the contribution of explicit memory to priming must ensure a short duration between prime and target.

The other aspect of experimental investigation which needs to consider the multi-system account of priming is the interpretation of experiment results. Different memory systems might vary in their contribution to priming for different kinds of material and in some cases might have opposing effects. Let us consider some hypothetical cases in order to clarify this point. If we assume that procedural memory is imple-

mented as an error-based system, then, as we discussed above, procedural memory will show larger priming after the DO when it is primed using the sequence PO—DO, as compared to when it is primed using the sequence DO—DO. In contrast, the long term memory in the priming system (as discussed in the case of model ①) will show a larger priming in the case of two consecutive DOs. Therefore, the two types of memory will have opposing contributions when measuring this effect. The experimenter must take these opposing effects into account when interpreting the results of the experiment. Similarly, when measuring the lexical boost, if the prime and target are separated by an intervening second prime that uses a different verb, but a competing syntactic structure, then the separate contributions from the long term and short term memories would mean that the priming would diminish, but the lexical boost should remain intact. Therefore, subjects will show greater priming when they are primed using the sequence [PO-Give]—[filler] and tested using [Give], than when they are primed using the sequence [PO-Give]—[DO-Send]. But since the lexical boost arises from a different memory system – one that did not have interference – it should be comparable in the two cases. These examples demonstrate the need to consider the contributions from different memory systems towards priming and not treat structural priming as a monolithic process.

§ 6.3.2.3. **Priming can be overcome.—** The final property of structural priming takes us back to our first model. While the other models chose to concentrate on the memory system, model ① investigated the influence of automaticity in the system on priming and lexical boost. This model showed that the amount of priming and lexical boost in the system is not fixed, but can vary as a result of the balance between local and nonlocal forces operating within the system.

The observation that structural priming is affected by both local and nonlocal constraints also helps us understand why structural priming shows properties of implicit learning – i.e. learning without awareness. According to Bargh (1994), awareness is one of the four components of automaticity (the others being intentionality, efficiency and controllability). A subject is likely to be unaware of automatic processes. The implicit learning literature is full of examples of experiments which show a dissociation between priming and recognition (see Schacter et al. (1993) for a review), suggesting subjects can be unaware of priming. Model ① explains why this might be the case. This model suggests that priming is due to hysteresis, which is the reluctance of a dynamical system to change its qualitative behaviour. Hysteresis is a local process and in

the absence of a nonlocal input, it ensures that the dynamical system will not change its stable state. Thus, the model argues that priming is due to inertia in the system and as such it is an automatic process. Being an automatic process, it does not need to be available to awareness.

Assuming that priming is an automatic process, model ① also showed that it can be overcome by nonlocal properties of the system. We grouped together such nonlocal properties into an 'arousal' module which controls the external input to each winner-take-all system. When the system is in a state of low arousal, it gives low external input to the WTA system, making it more inertial and leading to repetition. In the reverse case, when the arousal is high, the stable solutions of the system are completely governed by the external input and the system shows no inertia. In reality the arousal in the system will vary between these two extremes and the system will make decisions based on both automatic (local) and non-automatic (nonlocal) constraints. Bargh (1994) argued that automaticity is not a binary process and that there can be degrees of automaticity. Model ① implements these degrees of automaticity by controlling the balance between local and nonlocal processes through the level of arousal in the system.

Structural priming emerges as a robust phenomenon, repeatedly demonstrated in psychological experiments. But the amount of structural priming may vary not only based on the contribution from different memory systems (as we discussed above); here we argue that it may also vary based upon the balance between automatic and non-automatic cognitive processes. Many computational models such as the extended model considered in the last chapter or the error-based model developed by Chang et al. (2006) do not consider this critical balance between automatic and non-automatic processes and thus propagate the myth that structural priming is an unalterable phenomenon of human cognition. The case of the error-based model is particularly severe as it assumes that the same learning which is responsible for structural priming is also responsible for learning the syntactic rules of the language. This makes it impossible for the speaker to use the syntactic rules without being primed. In fact any account of priming which assumes that priming is due to (supervised or unsupervised) adjustment to procedural knowledge will struggle to show how a speaker can keep using procedural knowledge but still overcome priming.

Although experiments on lexical and semantic boost are a few experiments that systematically investigate the change in structural priming based on different features of an utterance, the topic requires much further investigation. Other features of dis-

course, such as awareness, audience design and discourse context might affect the degree of automaticity in the system and consequently also affect structural priming. Experiments need to be conducted which can investigate the role of such features in structural priming. These experiments will help to move this phenomenon from the domain of laboratory experiments to the domain of day-to-day conversations.

§ 6.3.2.4. **Predictions for future experiments.—** Throughout the thesis, we have made predictions for experimental studies based on insight gained from our theoretical and computational investigations. Here is a summary of the major predictions made during the thesis:

1. Based on the theory of priming due to error-based learning, developed in section 3.4, we predict that if learning is error-based then priming for consecutive trials of the same type should be lower than priming for consecutive trials of different types. [section 3.4.1.4]

2. Based on the same theory, we also predict that if learning is error-based, then the relative frequency of stimuli during the experiment should have an impact on the amount of priming. In particular, if the experiment consists of a larger number of primes of the lower frequency structure then priming should be larger than when the experiment consists of a larger number of primes of the higher frequency structure. [section 3.4.1.5]

3. Model ① predicts that the amount of priming depends on the level of automaticity (the *arousal*) in the system. When automaticity is high (and arousal is low), syntactic decisions are made locally and consequently priming is large. We predict that an experimental manipulation of automaticity in the cognitive system (e.g. through manipulation of a subject's awareness) can lead to change in the amount of priming. [section 4.6.1]

4. Model ① also predicts that the partial correlation between structural and lexical repetition, given the arousal in the system should be zero. That is, if we partial out the effect of arousal in the system, then lexical repetition should be uncorrelated to structural repetition. [section 4.6.1]

5. Model ② predicts that structural priming and lexical boost rely on different cognitive systems. Therefore it should be possible to find double dissociation between syntactic decision-making and the influence of lexical context on these

decisions. Thus, it should be possible to find individuals with short-term memory impairment but showing structural priming and equally, individuals with impaired syntactic processing should show a larger lexical boost than control subjects. [section 4.6.2]

6. Model ③ predicts that structural priming accumulates nonlinearly. Therefore, the amount of priming accrued over a series of trials will depend on the sequence in which the items are presented. [section 4.6.3]

7. The extended model decouples implicit learning from procedural memory and predicts that syntactic priming relies on implicit learning, but could be independent of learning of syntactic rules. Thus it separates syntactic acquisition from syntactic priming. In an experiment where subject try to acquire novel syntactic rules, this account predicts that it should be possible to decorrelate syntactic priming from syntactic acquisition. [section 6.2.1]

6.4 Future Work

We have presented four computational models in this thesis, starting with a simple model and moving towards more sophisticated models. We have also tried to explain a wide range of observations about structural priming and replicated the patterns of several experimental studies. Our final model has crossed over from the domain of structural priming to the domain of language production and has tried to explain how the cognitive processes during comprehension and production can lead to structural priming. But this success has come at a cost. We have filled logical gaps with computational and theoretical assumptions that demand justification. Some other assumptions require elaboration – perhaps into computational models that can form other subsystems of linguistic processing. There are also a number of other psychological experiments that we have not tried to replicate. Some of these experiments should be simple to model, but most would require us to extend and enrich the model. In this section, we describe various ways in which the work in this thesis can be taken forward. The list is not exhaustive, but identifies some of the key directions in which this work can be taken. We also make some speculative remarks about how to set out on each path.

§ 6.4.1 Theoretical extensions

First, we look at some of the computational assumptions that we made in order to implement the models and their alternatives. We also look at some natural extensions that would allow us to model more general principles of linguistic processing.

§ 6.4.1.1. **Binding mechanism.—** There are two binding mechanisms that we have implemented in this thesis. The first one, used in models ② and ③, binds nodes in the lexical and syntactic layers by allocating a node that stands for each conjunction between the two layers. The second binding mechanism, tensor-product binding, is an extension of the first mechanism for distributed representations. Both these mechanisms are *static binding* solutions – i.e. the nodes required to represent the bindings need to be preallocated.

We mentioned in Section 4.2.3 that preallocating nodes causes a waste of storage space. There will be certain nodes in the binding layer that will never be activated because the conjunction of patterns that they represent is never activated. This problem is particularly acute in a scheme like tensor-product binding which allocates a node for every possible combination. We saw that one alternative is to use something like holographic reduced representations (HRRs), but compressing information in this manner leads to noise and loss of information. We discussed that a particular problem we faced with these representations was that they required the vectors to be independent and identically distributed, a requirement that was in conflict with the requirements of the long term memory we were using.

A better solution to the problem of storage space is to use *dynamic binding*. Instead of preallocating nodes for each conjunction of nodes, this solution encodes the association between nodes by tagging them together. A tag could be any property of the system that can acquire at least two values. For the sake of illustration, let us say this property is the colour of a node and that the colour can be one of three different values – red, black or white. Let us also assume that nodes are white when they do not participate in any binding. When we want to associate two nodes, all we have to do is to paint them red or black. All nodes that are painted red will participate in one binding while all nodes that are painted black will participate in the other binding. This way, we can bind nodes together dynamically – i.e. without the need to specify a node for every combination of nodes.

Our simple solution of colouring the nodes does provide a binding mechanism, but a limited one. Firstly, it does not allow a node to participate in multiple bindings at

the same time. We can rectify this situation, by assuming that we do not colour the whole node, but instead just mark the node with a tiny spot. This way we can mark the node with both red and black spots allowing it to participate in both kinds of binding at the same time. The second problem with our simple solution is that it allows only two bindings – red and black. This problem can also be rectified easily by simply increasing the number of colours of the bindings. We can even choose a continuous variable as a tag instead of the discrete colour scheme, which will allow us to have a much larger (potentially infinite) number of tags. The last problem with the scheme is its neurological realisation: instead of assuming a colouring tag, we would like to replace it with a property that can be displayed by neurological systems. One popular choice for this property is the timing of neural firing (von der Malsburg, 1981; Gray & Singer, 1989; Shastri & Ajjanagadde, 1993). Physiological investigations show that neurons fire regularly at different frequencies. Therefore, two neurons can be tagged together if they fire in synchrony. Tagging nodes that fire synchronously makes sense because neural circuits can build a 'coincidence detector' that gets activated only when the two nodes fire at the same time (Kempter, Gerstner, Hemmel, & Wagner, 1998). Evidence for synchronous firing on neurons comes from measurement of neural activity in the cat's visual cortex which shows synchronous firing of neurons in response to presentation of visual stimuli (Singer & Gray, 1995).

To overcome the problem of spatial complexity, we would like to replace our static binding solution with a dynamic binding solution, such as binding through temporal synchrony. However, there are a number of computational details that will need to be worked out here. First of all, our current dynamical system assumes that the state of a node is simply determined by the value of its activation. In the neural parlance, this value of activation can be treated as the rate at which the neuron is firing. We will have to replace this rate code with a temporal code that specifies the precise timing of spikes in nodes. Furthermore, we will have to specify coincidence detectors that are temporarily tuned to detect specific synchronous firing. In order to specify the longevity of the binding, we will also have to specify how long this coincidence detection lasts, or when nodes lose their synchrony. Needless to say, each of these mechanisms will introduce complexity. However, they might also provide useful insight into the nature of decay in binding.

Of course, it is also possible that we do not use temporal synchrony to encode the binding between nodes at all and use another property of the system that allows us to tag representations together. However, this property will need to fulfill the three criteria

for a useful dynamic binding mentioned above: it should allow a node to participate in multiple bindings at the same time; it should allow a large number of bindings (equal to the number of conjunctive relations); it should be neurally plausible.

The dynamic binding scheme also helps us to remove an unwanted property of the tensor-product binding scheme. We noted above that the tensor product is wasteful because it preallocates all possible associations between nodes. This property of the tensor product also means that it replicates the information present in the associated modules. A tensor product between the syntax and schema modules represents all information represented in both the syntax and schema modules. From a computational perspective, this is obviously wasteful. Since the component patterns are stored in the syntax and schema modules anyway, all that the binding needs to store are pointers to the memories in these modules. The dynamic binding scheme gets rid of this profligacy by representing the binding by tagging the nodes themselves. In this case, the relationship between two modules can be stored in coincidence detectors which form pointers into the component patterns and represent the association much more efficiently.

A related point about the binding mechanism has to do with the way we have implemented the tensor-product binding. Currently we calculate this binding pair-wise between the three long term memory modules. However, one of the reasons why we chose to use the tensor-product binding and not just the outer product of two vectors was that tensors are generalisations of vectors that allow multi-dimensional indexing (Section 5.2.1.1). This means that tensor products allow us to bind representations in the three different modules (syntax, schema and lexical concepts) at the same time. Such binding would form a more accurate representation of an episode, which binds together lexical, syntactic and schema representations at the same time. However, the challenge is to perform this binding in such a manner that it preserves the structural properties of an utterance. In an utterance like *John gave Mary the book*, we want to bind the concept of *John* with the role of `GIVER` and the syntactic element *Subject*. Simply finding a tensor product between the three representations will not preserve this structural relationship. Ideally, the binding mechanism of the model should be extended in such a manner that it should allow us to store all the elements of the linguistic episode, whilst preserving the structural relationships.

§ 6.4.1.2. **Filler time.—** The filler trials in our simulations are modelled as a decay in memory between prime and target trials. But this is not how psychological experiments on structural priming implement fillers. Each filler consists of a comprehension

or production trial and uses a 'neutral' syntactic construction – i.e. a syntactic construction that is outside the cohort being tested. For example, Hartsuiker et al. (2008), who used the picture-naming task, used pictures which displayed actions that could be described with a *transitive* sentence. On the other hand, the prime and target trials used pictures that could be described using a *dative* sentence. The filler is 'neutral' in the sense that it does not overlap with the syntactic representations activated by the prime and target trials.

Now consider how we implement such a filler phase during our simulations. The prime and target trials in our experiment are separated by an adaptation phase. During this phase, memory decays along a time-based function. Usually an input file or input parameter specifies the time between prime and target. The shape of the function is usually an exponential curve. The choice of an exponential function is based on corpus analysis of how structural priming decays (Gries, 2005; Szmrecsanyi, 2006) and considerations of the nature of information retrieval from memory (J. R. Anderson, 1990). The rate of forgetting can be controlled by the time-constant for the exponential decay. The parameters specified at the beginning of the simulation set the rate of decay, while an input file provides the time of the filler phase.

If we want to accurately compare the results of the experiments with the results of the simulations, we have to calibrate the time and rate of decay in terms of a filler trial. While our current results provide a proof of concept, they do not make an accurate calibration, due to the lack of data available to perform such a calibration. One way to achieve better comparisons is to conduct psychological experiments which provide more information about these parameters. The memory literature is replete with experiments that study the nature of forgetting and the functions that can best describe this phenomenon (Wickelgren, 1970; Rubin, 1982; R. B. Anderson & Tweney, 1997). We believe that a similar enterprise for structural priming and lexical boost will be helpful. (The study conducted by Hartsuiker et al. (2008) is a step in the right direction).

But another way to obtain a better comparison between experiments and simulation is by modelling the effect of a 'neutral' filler on the representations in the network. We have assumed in our study that interference from the filler trials is manifested as an exponential forgetting curve. But a way to extend our work would be to look at how exactly a filler trial affects the retrieval of unrelated patterns presented during a prime trial. Frequently, memory literature assumes that this decay in memory is due to some cross-talk or interference between existing and newly stored patterns (an idea going back to (Müller & Pilzecker, 1900)), but it could also be due to competition

between representations (McGeoch, 1942), or perhaps purely a time-based decay (Page & Norris, 1998). These different possibilities are worth further investigation because they will provide a mechanistic explanation for the role of filler trials in psychological experiments. This reasoning also points to the possibility that different types of fillers are processed differently by the system and have different effects on the memory of the prime.

§ 6.4.1.3. **Syntactic knowledge.—** The models presented in this thesis have chosen to concentrate exclusively on the processes of memory that lead to grammatical encoding. We have looked at the memory of syntactic constructions themselves as well as the memory of the lexical concepts and the schema with which the construction was associated. However, grammatical encoding is a more complex decision. Memory of previous decisions plays only a part in grammatical encoding. Speakers use not only the memory of previous utterances, but also the rules of their grammatical knowledge in order to choose the grammatical form of an utterance. A more complete model of grammatical encoding should use a combination of memory and syntactic knowledge in order to achieve grammatical encoding.

The task of grammatical encoding can be broken down into a number of subprocesses. According to the model of language production proposed by Levelt (1989) and Bock and Levelt (1994), grammatical encoding consists of two stages: *functional processing* and *positional processing*. Functional processing itself consists of *lexical selection* and *function assignment*. It is this last process, that of function assignment, on which we have mostly concentrated.

A number of factors control function assignment. Levelt (1989) suggested that function assignment itself depends upon lexical selection – i.e. the process of function assignment is lexically driven. The key lexical item that provides the functional structure of the utterance is the verb (Bock & Levelt, 1994). In the extended model, it is the verb that provides the schema, or the frame, containing different roles. These roles need to be assigned a syntactic function and we have chosen to assign this syntactic function using the memory (stored in the tensor product) of the previous assignment. However, it would be naïve to suppose function assignment is done solely on the basis of previous memory. Clearly, our grammatical knowledge plays a major role in performing this function assignment. In fact, our grammatical knowledge should also constrain how function assignment is performed based on the memory of previous assignment. Evidence for the role of grammatical knowledge in sentence recall comes

from experiments performed by Lombardi and Potter (1992) which showed that the surface syntax of to-be-recalled sentence depended on the verb that was recalled. When subjects were lured into recalling an incorrect verb, with a different surface structure, they spontaneously adjusted the syntax of the recalled sentence to ensure grammaticality of the sentence. Thus the process of function assignment cannot depend completely on memory and needs to be constrained by the rules of grammar learnt by the subject. Our model, therefore, needs to include an additional pathway between the schema and syntax layers, which depends not on the memory of the association, but on the syntactic constraints provided by the schema (see Figure 6.4.1).

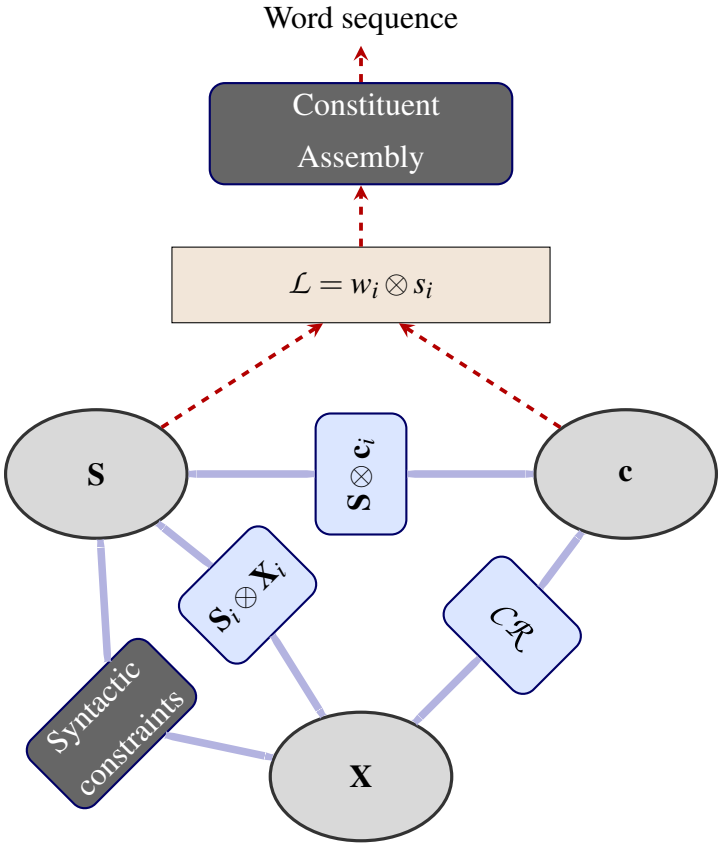


Figure 6.4.1: [Syntactic rules] Two additional modules are incorporated to account for the speaker's grammatical knowledge. One module takes care of the *syntactic constraints* of the verbs, while the other is responsible for *constituent assembly* for a list of function-marked words. For simplicity, the comprehension pathways are not shown.

The other place where the model needs to rely on grammatical knowledge is the stage of positional processing. This stage consists of two sub-processes: *constituent assembly* and *inflection*. The goal of our model is to provide correctly ordered words

and we will ignore whether or not each word is in the correctly inflected form. The process of constituent assembly, however, cannot be ignored. This process takes the list of words marked with their syntactic function and places them in a sequential order. So far, we have assumed that this knowledge is obtained from a lookup table which gives the correct order for every construction. However, we have reason to believe (Bock & Levelt, 1994) that constituent assembly takes places by hierarchically organising different constituents in accordance with the phrase structure rules of a language. But because this knowledge is abstract and independent of the function assignment process, we can specify a separate module in our model that lies between the long term memory of syntactic constructions and the output sequence of words.

Thus building in rules of procedural (grammatical) knowledge leads to the addition of two modules to our extended model (figure 6.4.1). The first module is parallel to the short term memory that binds the schema layer to the syntactic constructions. The second module lies between the syntactic constructions and the output sequence of words. Implementation of these additional modules will lead to a detailed account not only of structural priming, but also grammatical encoding.

§ 6.4.1.4. **Language acquisition.—** Related to the use of syntactic knowledge is its acquisition. Our model hypothesises that structural priming is due to a memory trace in the cognitive system. We have put forward this account in contrast to an error-based account of priming shown by Chang et al. (2006). This error-based account proposes that structural priming is due to learning in the procedural knowledge of the subject. We have discussed several drawbacks of such an error-based approach. However, a major advantage is that this model is more general in scope than ours and explains both structural priming and syntactic acquisition at the same time. Our model, on the other hand is silent on the processes of language acquisition.

The challenge here is to show how the speaker uses the same information to both participate in a linguistic activity (monologue or dialogue) and to acquire the rules of the language, at the same time. One way to think about the problem is the approach taken by the error-based model, which divides the system into two subsystems – one which stores the rules (through error-based learning) and the other which stores an episodic memory (not implemented by Chang et al. (2006)).

An alternative way of thinking about the problem is from an information processing perspective. We saw in Section 6.1 that language processing can be considered to be a transformation consisting of two kinds of processing – an *analysis* of infor-

mation so that it can be represented in a number of functionally specialised modules, and a *combination* of these representations. From this perspective, language acquisition consists of finding the right functions to perform this analysis and combination. In our model, we have assumed that we can represent input information in each of the three functionally specialised modules by looking up a connectionist representation for this input. Specifically, we have assumed a lookup table (ψ) that allows us to transform information backwards and forwards between a symbolic structure and its connectionist representation. One part of language acquisition is to understand how the system can learn to transform a linguistic signal into each of the functionally specialised representations. For example, learning the mapping from a sequence of words to the syntax module will involve learning the phrase structure rules that can derive the syntactic relations from word sequences. The other part of language acquisition comes from learning the combinations of the functionally specialised representations. For example, learning the binding between the schema and syntax representations leads to learning the syntactic constraints corresponding to each schema. Generalisation of different bindings leads to abstract rules of combining schema and syntax representations.

Thus, in order to implement language acquisition, the system will have to learn two principles of information transformation – analysis (or how the system learns to find a set of latent variables in information) and combination (or how it learns to find the functional relations that exist between the latent variables). In addition, a mechanistic account of language acquisition will have to show how the system *bootstraps* – i.e. how the system uses the principles of analysis and combination to determine which functional specialisations it should generate.

§ 6.4.2 Further simulations

Let us now look at some enhancements to our work that are suggested by existing experiments. We should be able to simulate some of these experiments with some minor changes to our models, while other experiments will require more careful planning and review of the models.

§ 6.4.2.1. **Semantic boost.—** Model ③ considered the effect of lexical representations on syntactic choice. In the last chapter, we extended this model so that we could also consider the flow of information from semantic representations (schema and lexical concepts) to syntactic constructions. Using this model allowed us to explain how

speakers might choose the syntactic structure of their utterances based on the schema that they wish to express. We saw that usually it is the verb that governs the structure of the schema and by studying the effect of the schema on the choice of syntax, we can actually explain the effect of repeating the verb on structural priming.

But expanding the model to the semantic domain gives us a much more powerful system that should allow us to test various other semantic influences on the choice of syntax. One study that measured such a semantic influence on syntactic choice was conducted by Cleland and Pickering (2003). This study used the picture-description paradigm to elicit picture descriptions in one of two orders: an adjective-noun order (*the red sheep*) or a noun-relative-clause order (*the sheep that's red*). In addition to observing structural priming and lexical boost, Cleland and Pickering (2003) also observed that semantic relatedness between prime and target enhanced priming. Participants were more likely to describe a picture as *the sheep that's red* after hearing *the goat that's red* than after hearing *the knife that's red*. They interpreted these results as showing a *semantic boost* because the semantic relatedness of the concepts SHEEP and GOAT led to an increase in priming as compared to the unrelated concepts SHEEP and KNIFE.

The extended model can capture such a semantic boost effect by analysing the role of connections between lexical concepts and syntactic structure. Currently, the short-term memory between these two layers does not play any role in the selection of syntactic structure during production. Moreover, different symbolic concepts are represented by linearly independent connectionist representations. Thus, currently the model will represent SHEEP and GOAT as two patterns of activation that do not overlap. In order to capture the semantic boost effect, we will have to find a connectionist representation that systematically encodes the semantic relationships between concepts.

One way to generate connectionist representations that encode semantic relationships between concepts is to use a feature map (Kohonen, 1990). A feature map represents semantic similarity between concepts as distance on the map. Words that are semantically similar are mapped close to each other while words that are different are farther away from each other. So far we have been generating the connectionist representations for each lexical concept using a lookup table (ψ). In order to represent semantic similarity, we can replace this lookup table with a feature map which takes the lexical symbol as an input and maps it onto a representation in semantic space.

However, implementing such a scheme in the current network will lead to some complexity in recall because of interference between different patterns. While the long

term memory does not assume completely independent representations, it is sensitive to the number of nodes activated by input stimulus. If too many nodes are activated, then the network has a chance of settling into either spurious or incorrect memories. Thus, if we want to implement the relationships between different concepts, we will have to overcome two challenges: (a) find the correct system of encoding these relationships, and (b) find a way to avoid spurious and incorrect retrievals due to interference.

§ 6.4.2.2. **Influence of primitive semantic features and thematic roles.—** A question related to the mechanism of semantic boost is the influence of other semantic and message level information on selection of syntactic structure. Three experimental studies are worth mentioning here. Bock and Loebell (1990) found that thematic role information is irrelevant to structural priming. However, Bock et al. (1992) found that primitive semantic features such as animacy influence the selection of syntactic functions and therefore do show structural priming. Finally, Chang et al. (2003) found in contrast to Bock and Loebell (1990), that subjects use thematic role information to assign syntactic functions when such an assignment did not alter the order of phrasal constituents (for details, see section 2.2.3.1).

We have already discussed in the previous chapter how the observations of Bock and Loebell (1990) are compatible with our model. The basic argument was that, in our representation scheme, thematic roles are local to each construction which meant that repeating the thematic role between prime and target would not influence syntactic priming, unless the thematic roles were part of the same construction. In order to replicate these results, we would need to extend the syntactic representation to encode locatives, in addition to datives. Otherwise, replication of these results should be fairly straightforward.

The other two studies, on the other hand, would require more careful consideration. Currently, our model does not have any representation of primitive semantic features, such as animacy. The extension of the model to include a feature map for encoding semantic representations should allow us to encode such semantic features. The next step would be to see how these semantic features are mapped onto different roles in a schema and then how these roles are mapped onto syntactic functions. The latter mapping is more straightforward as it is constrained by the syntactic construction. For example, a DO always maps the GIVER in the **GIVE** schema to the *Subject*. But the reasons for the former mapping – i.e. why the GIVER is more likely to be animate in

the target trial if it was animate in the prime trial, than when it was inanimate – will require a closer inspection.

Similarly, simulating the experiment of Chang et al. (2003) will also require further development. As noted above, our model assumes that the thematic role assignments are local to each construction (schema). The influence of thematic roles on function assignment means that there is some connection between different schemata, based on thematic role overlap. Much like the representation of lexical concepts, the connectionist representations of different schemata are linearly independent in the extended model – i.e. there is no overlap between them. But, as suggested by construction grammar (Fillmore & Kay, 1993; Langacker, 1987), different constructions are not stored as an unstructured list, but hierarchically organised into a taxonomic network. Different constructions are related to each other in such a network through a relationship of schematicity or generality, with more general constructions occupying higher nodes in the network (Croft & Cruse, 2004). Such a hierarchical organisation could permit an overlap between different constructions that share thematic roles. However, adopting such organisation must also explain why the influence of thematic roles on function assignment is a weak one and only manifests itself when the two assignments share the order of phrasal constituents, as illustrated by Chang et al. (2003). A number of different explanations is possible, including the fact that the difference in priming could be due to a difference of focus in the two types of primes rather than a difference in the thematic roles.

§ 6.4.2.3. **Production-to-production priming.—** We have mentioned in the previous section that our model should be able to show production-to-production priming. However, all the simulations considered in the last two chapters involve comprehension-to-production priming. The model learns in the same manner during comprehension and production – by retrieving long-term memories and storing bindings. In this sense, the production trial leaves as much of a mark upon the system as the comprehension trial. However, in order to demonstrate this, we must design a simulation that will allow the model to show production-to-production priming.

The problem here is that the design of such a simulation is not completely obvious. First of all, if the prime trial is a production trial, then the model can produce any of the alternative structures during the prime. This makes it difficult to ensure that both the structures will be produced roughly an equal number of times, so that we can compare the amount of priming for each structure. Secondly, if the simulation contains

only production trials, then the effect of structural priming will mean that it will tend to produce the structure that it had produced in the previous trial. Thus, as the experiment goes on, one of the structures will become more and more likely to be produced. In fact, in the absence of noise, the model will get stuck producing only a single kind of structure.

One possible design involves having comprehension trials at regular intervals during the experiment, so that the model can 'unlearn' some of the priming incurred during the production trials, making the alternative structures more balanced. While this solution seems feasible, it is not ideal as it will become difficult to separate the effect of priming from production and comprehension trials. Moreover, there do not seem to be any psychological experiments available to which we can compare the results from this simulation.

6.5 Final remarks

In this chapter, we have broadened the discussion of structural priming to other aspects of human cognition, including its ability to perform functional specialization and integration. We have seen how such functional specialization agrees with the architecture of the models presented in this thesis and helps to explain the properties of structural priming. We have focused on the temporal properties of priming and shown how structural priming and lexical boost can show different rates of decay. The key to understanding the difference in structural priming and lexical boost lies in how these properties are encoded in our models. While structural priming is due to a module responsible for functional specialization, lexical boost is due to a module responsible for integrating (binding) such specialized representations. We have also discussed a multi-system approach to structural priming, arguing that structural priming could be a result of learning in different kinds of memory in the cognitive system. While treating priming as a consequence of different types of memories brings the study of priming closer to the study of human memory systems, it also highlights the limitations of our models. We have enumerated several possible ways in which this study can be taken forward and the models can be extended to account for data available from other psychological studies.

This thesis develops a series of mathematical models that enhance our understanding of language production, in general, and structural priming, in particular. These models present a detailed account of the learning mechanisms that could be responsi-

ble for structural priming and move the debate about the existence of structural priming from a functional domain (whether it is due to implicit or explicit learning) to a computational domain (which learning algorithms are responsible for structural priming). These models also provide the novel framework of dynamical systems for the study of priming and memory. Concepts such as hysteresis, adaptation and winner-take-all competition give a natural way of capturing cognitive processes that may underlie language production and lead to the temporal properties of structural priming.

In this thesis, we have reviewed the importance of studying structural priming and seen the limitations of an error-based account in explaining some properties and patterns of priming. Our first model demonstrated how dynamical systems can be used to implement structural priming and also showed how a balance between local and nonlocal properties can lead to lexical boost. The second and third models provided a detailed temporal analysis of the memory of an utterance and distinguished three different types of learning that could lead to structural priming: hysteresis in mutually excitatory binding nodes, hysteresis in mutually inhibitory winner-take-all nodes and incremental learning of input sensitivity. Using these mechanisms, these models were able to account for the results from several psychological studies. The extended model took this work further and overcame several computational and representational limitations of previous models. By simulating this model, we showed how structural priming may rely on different memory systems which are accessed during comprehension and production. We also showed how some processes and memory systems could overlap between comprehension and production and hence how linguistic processing during one could be intricately tied to linguistic processing in the other.

A

GUI

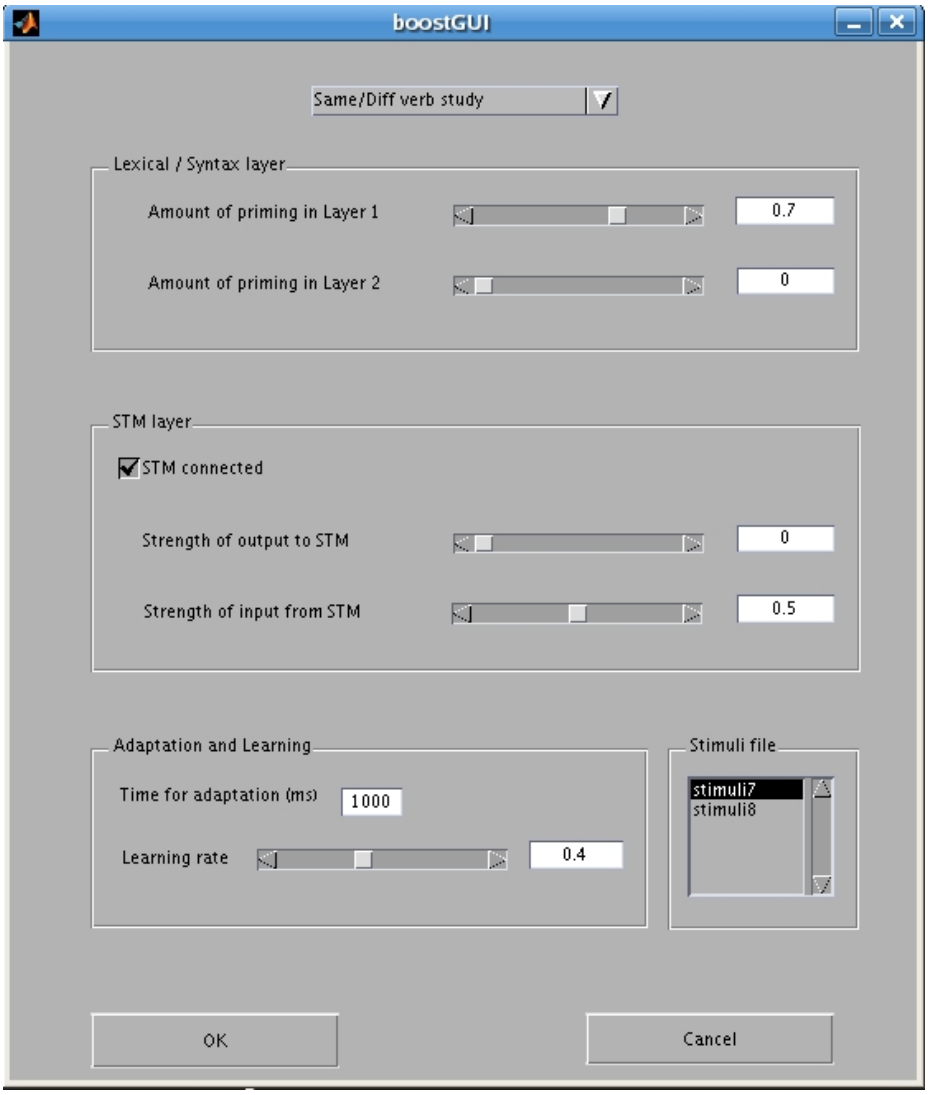


Figure A.0.1: The graphical user interface (GUI) used for testing model ③. The drop-down menues and textboxes are used to control the free parameters of the model.

B

Stimuli file

Part of a stimuli file used to test model ③ is shown here. The file consists of blocks of 16 lines, where each block is used to train a particular subject. Each block consists of two phases – the training phase and the testing phase. The training phase is made of the first 10 episodes and the testing phase is made of the last six. Each episode consists of prime and target trials. The prime trial specifies both the verb and the syntactic construction (i.e. simulates comprehension), while the target trial specifies only the verb. The target trials during the training phase do not specify any verb and allow the network to choose a verb randomly.

```
    // Subject 1
Episode1:  Prime[DO][Lend] Target[0]
Episode2:  Prime[PO][Lend] Target[0]
Episode3:  Prime[PO][Hand] Target[0]
Episode4:  Prime[PO][Hand] Target[0]
Episode5:  Prime[PO][Lend] Target[0]
Episode6:  Prime[DO][Hand] Target[0]
Episode7:  Prime[PO][Lend] Target[0]
Episode8:  Prime[DO][Lend] Target[0]
Episode9:  Prime[DO][Lend] Target[0]
Episode10:  Prime[DO][Lend] Target[0]
Episode11:  Prime[PO][Lend] Target[Lend]
Episode12:  Prime[PO][Hand] Target[Hand]
Episode13:  Prime[DO][Lend] Target[Lend]
Episode14:  Prime[PO][Hand] Target[Hand]
Episode15:  Prime[DO][Hand] Target[Hand]
Episode16:  Prime[DO][Hand] Target[Hand]
```

```
    // Subject 2
Episode1:  Prime[PO][Give] Target[0]
Episode2:  Prime[DO][Send] Target[0]
Episode3:  Prime[PO][Give] Target[0]
Episode4:  Prime[PO][Give] Target[0]
Episode5:  Prime[DO][Give] Target[0]
Episode6:  Prime[PO][Send] Target[0]
Episode7:  Prime[DO][Give] Target[0]
Episode8:  Prime[DO][Send] Target[0]
Episode9:  Prime[DO][Send] Target[0]
Episode10:  Prime[PO][Give] Target[0]
Episode11:  Prime[DO][Give] Target[Give]
Episode12:  Prime[PO][Give] Target[Give]
Episode13:  Prime[PO][Send] Target[Send]
Episode14:  Prime[PO][Give] Target[Give]
Episode15:  Prime[DO][Give] Target[Give]
Episode16:  Prime[DO][Send] Target[Send]

   ...
```

References

Amari, S. (1977). Dynamics of pattern formation in lateral inhibition type neural fields. *Biological Cybernetics*, *27*, 77–87.

Anderson, J. R. (1983). A spreading activation theory of memory. *Journal of verbal learning and verbal behavior*, *22*, 261–295.

Anderson, J. R. (1990). *The adaptive character of thought*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Anderson, J. R. (1993). *Rules of the mind*. Hillsdale, NJ: Earlbaum.

Anderson, J. R., Budiu, R., & Reder, L. M. (2001). A theory of sentence memory as part of a general theory of memory. *Journal of Memory and Language*, *45*, 337–367.

Anderson, J. R., & Schooler, L. J. (1991). Reflections of the environment in memory. *Psychological Science*, *2*, 396–408.

Anderson, R. B., & Tweney, R. B. (1997). Artifactual power curves in forgetting. *Memory and Cognition*, *25*(5), 724–730.

Baddeley, A. D. (2000). The episodic buffer: a new component of working memory. *Trends in Cognitive Science*, *4*(11), 417–423.

Baddeley, A. D., & Hitch, G. (1974). Working memory. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory.* New York: Academic Press.

Bargh, J. A. (1994). The four horsemen of automaticity: Awareness, intension, efficiency and control in social cognition. In R. S. Wyer & T. K. Scrull (Eds.), *Handbook of social cognition* (Vol. 1, pp. 1–40). Hillsdale, NJ: Lawrence Earlbaum Associates, Inc.

Bates, E., & Goodman, J. C. (2001). On the inseparability of grammar and the lexi-

con: Evidence from acquisition. In M. Tomasello & E. Bates (Eds.), *Language development: The essential readings* (pp. 134–162). Oxford, England: Basil Blackwell.

Becker, S. (1995). Unsupervised learning with global objection functions. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 997–1000). MIT Press.

Bell, A. J., & Sejnowski, T. J. (1995). An information maximization approach to blind-separation and blind deconvolution. *Neural Computation*, *7*, 1129–1159.

Bock, K. (1986). Syntactic persistence in language production. *Cognitive Psychology*, *18*, 355–387.

Bock, K. (1989). Closed class immanence in sentence production. *Cognition*, *31*, 163–186.

Bock, K., & Cutting, J. C. (1992). Regulating mental energy: Performance units in language production. *Journal of Memory and Language*, *31*, 99–127.

Bock, K., Dell, G. S., Chang, F., & Onishi, K. H. (2007). Persistent structural priming from language comprehension to language production. *Cognition*, *104*, 437–458.

Bock, K., & Griffin, Z. M. (2000). The persistence of structural priming: Transient activation or implicit learning? *Journal of Experimental Psychology, General*, *129*, 177–192.

Bock, K., & Levelt, W. (1994). Language production: Grammatical encoding. In M. A. Gernsbacher (Ed.), *Handbook of psycholinguistics* (pp. 945–984). San Diego: Academic Press.

Bock, K., & Loebell, H. (1990). Framing sentences. *Cognition*, *35*, 1–39.

Bock, K., Loebell, H., & Morey, R. (1992). From conceptual roles to structural relations: Bridging the synaptic cleft. *Psychological Review*, *99*, 150–171.

Branigan, H. P., McLean, J. F., Thatcher, K., & Jones, M. W. (2006). A blue cat or a cat that is blue? abstract syntax in young children's noun phrase production. London.

Branigan, H. P., Pickering, M. J., & Cleland, A. (1999). Syntactic priming in written production: Evidence for rapid decay. *Psychonomic Bulletin and Review*, *6*, 635–640.

Branigan, H. P., Pickering, M. J., & Cleland, A. A. (2000). Syntactic co-ordination in dialogue. *Cognition*, *75*, B13–B25.

Brennan, S. E., & Clark, H. H. (1996). Conceptual pacts and lexical choice in conver-

sation. *Journal of Experimental Psychology: Language, Memory and Cognition*, *22*, 1482–1493.

Brennan, S. E., & Metzing, C. A. (2004). Two steps forward, one step back: Partner-specific effects in a psychology of dialogue. *Behavioral and Brain Sciences*, *27*, 192–193.

Brown, R., & McNeill, D. (1966). The "tip-of-the-tongue" phenomenon. *Journal of Verbal Learning and Verbal Behavior*, *5*, 325–337.

Carandini, M., Demb, J. B., Mante, V., Tolhurst, D. J., Dan, Y., Olshausen, B. A., et al. (2005). Do we know what the early visual system does? *The Journal of Neuroscience*, *25*(46), 10577–10597.

Carlson, R. A., & Dulany, D. E. (1985). Conscious attention and abstraction in concept learning. *Journal of Experimental Psychology: Learning, Memory and Attention*, *11*, 45–58.

Chang, F., Bock, K., & Goldberg, A. (2003). Can thematic roles leave traces of their places? *Cognition*, *90*, 29–49.

Chang, F., Dell, G. S., & Bock, K. (2006). Becoming syntactic. *Psychological Review*, *113*(2), 234–272.

Chang, F., Dell, G. S., Bock, K. J., & Griffin, Z. M. (2000). Structural priming as implicit learning: A comparison of models of sentence production. *Journal of Psycholinguistic Research*, *29*, 217–229.

Clark, H. H. (1996). *Using language*. Cambridge: Cambridge University Press.

Cleland, A. A., & Pickering, M. J. (2003). The use of lexical and syntactic information in language production: Evidence from the priming of noun-phrase structure. *Journal of Memory and Language*, *49*, 214–230.

Cleland, A. A., & Pickering, M. J. (2006). Do writing and speaking employ the same syntactic representations? *Journal of Memory and Language*, *54*, 185–198.

Collins, A. M., & Loftus, E. F. (1975). A spreading-activation theory of semantic processing. *Psychological Review*, *82*(6), 407–428.

Corley, M., & Scheepers, C. (2002). Syntactic priming in english sentence production. *Psychonomic Bulletin and Review*, *9*, 126–131.

Croft, W., & Cruse, D. A. (2004). *Cognitive linguistics*. Cambridge, UK: Cambridge University Press.

Dayan, P., & Abbott, L. F. (2001). *Theoretical neuroscience*. Cambridge, MA: MIT Press.

Dell, G. S. (1986). A spreading activation theory of retrieval in sentence production.

Psychological Review, *93*(3), 283–321.

Eichenbaum, H. (2003). How does hippocampus contribute to memory? *Trends in Cognitive Science*, *7*(10), 427–429.

Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, *14*, 179–211.

Ferreira, V. S. (2005). Maintaining syntactic diversity: Syntactic persistence, the inverse-preference effect, and syntactic affirmative action. *Unpublished manuscript*.

Ferreira, V. S., & Bock, K. (2006). The functions of structural priming. *Language and Cognitive Processes*, *21*, 1011–1029.

Ferreira, V. S., Bock, K., Wilson, M. P., & Cohen, N. J. (2008). Memory for syntax despite amnesia. *Psychological Science*, *19*(9), 940–946.

Fillmore, C. J. (1985). Frames and the semantics of understanding. *Quaderni di semantica*, *6*, 222–254.

Fillmore, C. J., & Atkins, B. T. (1992). Toward a frame-based lexicaon: the semantics of risk and its neighbors. In A. Lehrer & E. F. Kittay (Eds.), *Frames, fields and contrasts: new essays in semantic and lexical organization.* Lawrence Erlbaum Associates.

Fillmore, C. J., & Kay, P. (1993). *Construction grammar coursebook*. Berkeley, CA: University of California.

Fisher, C. (2002). The role of abstract syntactic knowledge in language acquisition: A reply to tomasello. *Cognition*, *82*, 259–278.

Fodor, J. A. (1983). *Modularity of mind: An essay on faculty psychology*. Cambridge, MA: MIT Press.

Fodor, J. A., & Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: a critical analysis. In S. Pinker & J. Mehler (Eds.), *Connections and symbols* (pp. 3–71). Cambridge, MA: MIT Press.

Friston, K. (2005). A theory of cortical responses. *Philosophical Transactions of The Royal Society B*, *360*, 815–836.

Fromkin, V. A. (1971). The non-anomalous nature of anomalous utterances. *Language*, *47*(1), 27–52.

Funahashi, S., Bruce, C. J., & Goldman-Rakic, P. S. (1989). Mnemonic coding of visual space in the monkey's dorsolateral prefrontal cortex. *Journal of Neurophysiology*, *61*(2), 331–349.

Garrett, M. F. (1975). The analysis of sentence production. In G. H. Bower (Ed.), *The psychology of learning and motivation* (Vol. 9, pp. 133–177). New York:

Academic Press.

Garrett, M. F. (1980). Levels of processing in sentence production. In B. Butterworth (Ed.), *Language production* (Vol. 1, pp. 177–220). London: Academic Press.

Garrett, M. F. (1988). Processes in language production. In F. J. Newmeyer (Ed.), *Linguistics: The cambridge survey: Iii. language: Psychological and biological aspects* (pp. 69–96). Cambridge: Cambridge University Press.

Garrod, S., & Anderson, A. (1987). Saying what you mean in dialogue: A study in conceptual and semantic co-ordination. *Cognition*, *27*, 181–218.

Garrod, S., & Pickering, M. (2007). Automaticity in language production in monologue and dialogue. In A. S. Meyer, L. R. Wheeldon, & A. Krott (Eds.), *Automaticity and control in language processing* (pp. 1–21). Hove:Psychology Press.

Garrod, S., & Pickering, M. J. (2004). Why is conversation so easy? *Trends in Cognitive Science*, *8*, 8–11.

Garrod, S., & Pickering, M. J. (in press). Alignment in dialogue. In G. Gaskell (Ed.), *Oxford handbook of psycholinguistics.* Oxford: Oxford University Press.

Gleitman, L. (1990). The structural sources of verb meanings. *Language Acquisition: A Journal of Developmental Linguistics*, *1*, 3–55.

Goldinger, S. D. (1998). Echoes of echoes? an episodic theory of lexical access. *Psychological Review*, *105*(2), 251–279.

Gray, C. M., & Singer, W. (1989). Stimulus specific neuronal oscillations in orientationcolumns of cat visual cortex. *Proceedings of the National Academy of Sciences USA*, *86*, 1698–1702.

Grice, H. P. (1975). Logic and conversation. In P. Cole & J. Morgan (Eds.), *Syntax and semantics 3: Speech acts.* New York: Academic Press.

Gries, S. T. (2005). Syntactic priming: a corpus-based approach. *Journal of Psycholoinguistic Research*, *34*(4), 365–399.

Griffin, Z. M., & Weinstein-Tull, J. (2003). Conceptual structure modulates structural priming in the production of complex sentences. *Journal of Memory and Language*, *49*, 537–555.

Grill-Spector, K., Henson, R., & Martin, A. (2006). Repetition and the brain: neural models of stimulus-specific effects. *Trends in Cognitive Science*, *10*(1), 14–23.

Grush, R. (2004). The emulation theory of representation: motor control, imagery, and perception. *The Behavioural and Brain Sciences*, *27*, 377–435.

Hale, J. (2006). Uncertainty about the rest of the sentence. *Cognitive Science*, *30*(4), 609–642.

Hamann, S. B., & Squire, L. R. (1997). Intact perceptual memory in the absence of conscious memory. *Behavioral Neuroscience*, *111*, 850–854.

Harley, T. A. (2001). *The psychology of language* (2nd ed.). Psychology Press.

Hartsuiker, R. J., Bernolet, S., Schoonbaert, S., Speybroeck, S., & Vanderelst, D. (2008). Syntactic priming persists while the lexical boost decays: Evidence from written and spoken dialogue. *Journal of Memory and Language*, *58*(2), 214–238.

Hartsuiker, R. J., Kolk, H. H. J., & Huiskamp, P. (1999). Priming word order in sentence production. *Journal of Experimental Psychology*, *52A*, 129–147.

Hartsuiker, R. J., & Westenberg, C. (2000). Word order priming in written and spoken sentence production. *Cognition*, *75*, B27–B39.

Haykin, S. (1999). *Neural networks – a comprehensive foundation*. Upper Saddle River, NJ: Prentice-Hall Inc.

Henson, R., Shallice, T., & Dolan, R. (2000). Neuroimaging evidence for dissociable forms of repetition priming. *Science*, *287*, 1269–1272.

Hertz, J., Krogh, A., & Palmer, R. (1991). *Introduction to the theory of neural computation*. Reading, MA: Addison-Wesley Publishing Company.

Hinton, G. E. (1990). Mapping part-whole hierachies into connectionist netowrks. *Artificial Intelligence*, *46*, 47–76.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1986). Distributed representations. In D. E. Rumelhart & J. L. McClelland (Eds.), *Paralled distributed processing: Explorations in the microstructure of cognition* (Vol. 1, pp. 77–109). Cambridge, MA: MIT Press.

Hodgkin, A., & Huxley, A. (1952). A quantitatie descriptionof membrane current and its application to conduction and excitation in nerve. *Journal of Physiology*, *117*, 500–544.

Hoppensteadt, F. C., & Izhikevich, E. M. (1997). *Weakly connected neural networks*. New York: Springer-Verlag.

Horton, W. S., & Keysar, B. (1996). When do speakers take into account common ground? *Cognition*, *59*(1), 91–117.

Hubel, D. H., & Wiesel, T. N. (1962). Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, *160*, 106–154.

Hummel, J. E., & Biederman, I. (1992). Dynamic binding in a neural network for shape recognition. *Psychological Review*, *99*, 480–517.

Kaschak, M. P., & Borreggine, K. L. (2008). Is long-term structural priming affected by patterns of experience with individual verbs? *Journal of Memory and Language*, *58*(3), 862–878.

Kay, P., & Fillmore, C. J. (1999). Grammatical constructions and linguistic generalizations: the what's X doing Y? construction. *Language*, *75*, 1–33.

Kempen, G., & Hoenkamp, E. (1987). An incremental procedural grammar for sentence formulation. *Cognitive Science*, *11*, 201–258.

Kempter, R., Gerstner, W., Hemmel, J. L. van, & Wagner, H. (1998). Extracting oscillations: neuronal coincidence detection with noisy periodic spike input. *Neural Computation*, *10*, 1987–2017.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*, 1464–1480.

Langacker, R. W. (1987). *Foundations of cognitive grammar*. Stanford, CA: Stanford University Press.

Levelt, W. J. M. (1989). *Speaking: From intention to articulation*. Cambridge, MA: MIT Press.

Levelt, W. J. M., & Kelter, S. (1982). Surface form and memory in question answering. *Cognitive Psychology*, *14*, 78–106.

Levelt, W. J. M., Roelofs, A., & Meyer, A. S. (1999). A theory of lexical access in speech production. *Behavioral and Brain Sciences*, *22*, 1–75.

Lieven, E. V. M., Behrens, H., Speares, J., & Tomasello, M. (2003). Early syntactic creativity: A usage-based approach. *Journal of Child Language*, *30*, 333–367.

Lombardi, L., & Potter, M. C. (1992). The regeneration of syntax in short term memory. *Journal of Memory and Language*, *31*, 713–733.

MacWhinney, B. (1987). The competition model. In B. MacWhinney (Ed.), *Mechanisms of language acquisition* (pp. 249–308). Hillsdale, NJ: Earlbaum.

McClelland, J. L., McNaughton, B. L., & O'Reilly, R. C. (1995). Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory. *Psychological Review*, *102*(3), 419–457.

McGeoch, J. A. (1942). *The psychology of human learning: An introduction*. New York: Longmans.

Mendel, J. M., & McLaren, R. W. (1970). Reinforcement learning control and pattern recognition systems. In J. M. Mendel & K. S. Fu (Eds.), *Adaptive learning and pattern recognition systems: Theory and applications* (pp. 287–318). Academic

Press.

Meringer, R., & Mayer, K. (1993). *Versprechen und verlesen*. Stuttgart: Goschensche Verlag.

Miller, E. K., & Desimone, R. (1994). Parallel neuronal mechanisms for short-term memory. *Science*, *263*, 520–522.

Milner, B. (1962). Les troubles de la memoire accompagnant des lesions hippocampiques bilaterales. In B. Milner & S. Glickman (Eds.), *Psychologie de l'hippocampe.* Paris: Centre National de la Recherche Scientifique.

Müller, G. E., & Pilzecker, A. (1900). Experimentalle beitrage zur lehre com gedachtnis. *Zeitschrift fur Psychologie*, *1*, 1–300.

Naigles, L. R. (2002). Form is easy, meaning is hard: Resolving a paradox in early child language. *Cognition*, *86*, 157–199.

Naka, K. I., & Rushton, W. A. (1966). S-potentials from colour units in the retina of fish. *J. Physiol.*, *185*, 584–599.

Oja, E. (1982). A simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, *15*, 267–273.

O'Keefe, J., & Nadel, L. (1978). *The hippocampus as a cognitive map*. Oxford, UK: Oxford University Press.

Page, M. P. A., & Norris, D. (1998). The primacy model: A new model of immediate serial recall. *Psychological Review*, *105*(4), 761–781.

Pickering, M. J., & Branigan, H. P. (1998). The representation of verbs: Evidence from syntactic priming in language production. *Journal of Memory and Language*, *39*, 633–651.

Pickering, M. J., & Ferreira, V. S. (2008). Structural priming: A critical review. *Psychological Bulletin*, *134*(3), 427–459.

Pickering, M. J., & Garrod, S. (2004). Toward a mechanistic psychology of dialogue. *Behavioral and Brain Sciences*, *27*, 169–226.

Pickering, M. J., & Garrod, S. (2007). Do people use language production to make predictions during comprehension? *Trends in Cognitive Science*, *11*, 105–110.

Plate, T. A. (1995). Holographic reduced representations. *IEEE Transactions on Neural Networks*, *6*(3), 623–641.

Plate, T. A. (1997). A common framework for distributed representation schemes of compositional structure. In F. Maire, R. Hayward, & J. Diedrich (Eds.), *Connectionist systems for knowledge representation and deduction.* Queensland University of Technology.

Pollack, J. B. (1990). Recursive distributed representations. *Artificial Intelligence*, *46*(1-2), 77–105.

Posner, M. I., & Petersen, S. E. (1990). The attentional system of the human brain. *Annual Review of Neuroscience*, *13*, 25–42.

Quillian, M. R. (1962). A revised design for an understanding machine. *Mechanical Translation*, *7*, 17–29.

Quillian, M. R. (1967). Word concepts: A theory and simulation of some basic semantic capabilities. *Behavioral Science*, *12*, 410–430.

Reber, A. S. (1967). Implicit learning of artificial grammar. *Journal of verbal learning and verbal behaviour*, *6*, 855–863.

Reitter, D. (2008). *Context effects in language production: Models of syntactic priming in dialogue corpora*. Unpublished doctoral dissertation, School of Informatics, University of Edinburgh.

Roelofs, A. (1992). A spreading activation theory of lemma retrieval in speaking. *Cognition*, *42*, 107–142.

Roelofs, A. (1993). Testing a non-decompositional theory of lemma retrieval in speaking: Retrieval of verbs. *Cognition*, *47*, 59–87.

Rubin, D. C. (1982). On the retention function for autobiographical memory. *Journal of Verbal Learning and Verbal Behaviour*, *21*, 21–38.

Rumelhart, D. E., Smolensky, P., McClelland, J. L., & Hinton, G. E. (1986). Schemata and sequential thought processes in pdp models. In J. L. McClelland & D. E. Rumelhart (Eds.), *Parallel distributed processing* (Vol. 2, pp. 7–57). Cambridge, MA: MIT Press.

Sandberg, A., Lansner, A., Petersson, K.-M., & Ekeberg, O. (2000). A palimpsest memory based on an incremental bayesian learning rule. *Neurocomputing*, *32-33*, 987–994.

Schacter, D. L., Chiu, C. Y. P., & Ochsner, K. N. (1993). Implicit memory: A selective review. *Annual Review of Neuroscience*, *16*, 159–182.

Scheepers, C. (2003). Syntactic priming of relative clause attachments: persistence of structural configuration in sentence production. *Cognition*, *89*, 179–205.

Schoonbaert, S., Hartsuiker, R. J., & Pickering, M. J. (2007). The representation of lexical and syntactic information in bilinguals: Evidence from syntactic priming. *Journal of Memory and Language*, *56*, 153–171.

Senn, W., & Fusi, S. (2005). Learning only when necessary: Better memories for correlated patterns in networks with bounded synapses. *Neural Computation*,

17, 2106–2138.

Seydel, R. (Ed.). (1999). *World of bifurcation.* Online Collection and Tutorials of Nonlinear Phenomena.

Shastri, L., & Ajjanagadde, V. (1993). From simple associations to systematic reasoning: a connectionist representation of rules, variables and dynamic bindings. *Behavioral and Brain Sciences*, *16*, 417–494.

Sikstrom, S. (1999). Power function forgetting curves as an emergent property of biologically plausible neural network models. *International Journal of Psychology*, *34*, 460–464.

Sikstrom, S. (2002). Forgetting curves: Implications for connectionist models. *Cognitive Psychology*, *45*, 95–152.

Singer, W., & Gray, C. M. (1995). Visual feature integration and the temporal correlation hypothesis. *Annual Review of Neuroscience*, *18*, 555–586.

Smolensky, P. (1987). *On variable binding and the representation of symbolic structure in connectionist systems. technical report*. Bolder, CO: University of Colorado.

Smolensky, P. (1990). Tensor product variabe binding and the representation of symbolic structures in connectionist systems. *Artificial Intelligence*, *46*, 159–216.

Squire, L. R. (1992). Memory and the hippocampus: A synthesis from findings with rats, monkeys, and humans. *Psychological Review*, *99*(2), 195–231.

Squire, L. R. (2004). Memory systems of the brain: A brief history and current perspective. *Neurobiology of learning and memory*, *82*, 171–177.

Steedman, M. (2000). *The syntactic process*. Cambridge, MA: MIT Press.

Szmrecsanyi, B. (2006). *Morphosyntactic persistence in spoken english: a corpus study at the intersection of variationist sociolinguistics, psycholinguistics, and discourse analysis*. KG, Berlin: Mouton de Gruyter.

Talati, A., & Hirsch, J. (2005). Functional specialization within the medial frontal gyrus for perceptual Go/No-Go decisions based on "What," "When" and "Where" related information: An fMRI study. *Journal of Cognitive Neuroscience*, *17*(7), 981–993.

Thorpe, S. (1995). Localized versus distributed representations. In M. A. Arbib (Ed.), *The handbook of brain theory and neural networks* (pp. 549–552). Cambridge, MA: MIT Press.

Tian, B., Reser, D., Durham, A., Kustov, A., & Rauschecker, J. P. (2001). Functional specialization in rhesus monkey auditory cortex. *Science*, *292*(5515), 290–293.

Tomasello, M. (1992). *First verbs: A case study of early grammatical development*. Cambridge, England: Cambridge University Press.

Tomasello, M. (2003). *Constructing a language: A usage-based theory of language acquisition*. Cambridge, MA: Harvard University Press.

Tulving, E. (1984). Precis of elements of episodic memory. *The Behavioural and Brain Sciences*, *7*, 223–268.

Tulving, E. (1995). Organization of memory: Quo vadis? In M. Gazzaniga (Ed.), *The cognitive neurosciences* (pp. 839–847).

Tulving, E. (2007). Are there 256 kinds of memory? In *The foundations of remembering: Essays in honour of henry l. roediger, iii.* New York, NY: Psychology Press.

Tulving, E., Hayman, C. A. G., & Macdonald, C. (1991). Long lasting perceptual priming and semantic learning in amnesia: A case experiment. *Journal of Experimental Psycholgy: Learning, Memory and Cognition*, *8*, 352–373.

Tulving, E., & Schacter, D. L. (1990). Priming and human memory systems. *Science*, *247*, 301–306.

Vigliocco, G., Antonini, T., & Garrett, M. F. (1997). Grammatical gender is on the tip of italian tongues. *Psychological Science*, *8*(4), 314–317.

von der Malsburg, C. (1981). The correlation theory of brain function. *Internal report, Deptartment of Neurobiology, Max-Planck-Institute for Biophysical Chemistry*, *81*.

Wickelgren, W. A. (1970). Time, interference, and rate of presentation in short-term recognition memory for items. *Journal of Mathematical Psychology*, *7*, 219–235.

Willshaw, D. (1981). Holography, associative memory and inductive generalization. In G. E. Hinton & J. E. Anderson (Eds.), *Parallel models of associative memory.* Hillsdale.

Wilson, B. A., & Baddeley, A. D. (1988). Semantic, episodic and episodic memory in a post-meningitic amnesic patient. *Brain and Cognition*, *8*, 31–46.

Wilson, H. R. (1999). *Spikes, decisions and actions*. Great Clarendon Street, Oxford: Oxford University Press.

Wilson, H. R., & Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophysical Journal*, *12*, 1–24.

Wilson, H. R., & Humanski, R. (1992). Spatial frequency adaptation and contrast gain control. *Vision Research*, *33*(8), 1133–1149.

Wilson, M., & Knoblich, G. (2005). The case for motor involvement in perceiving conspecifics. *Psychological Bulletin*, *131*, 460–473.

Zeki, S., Watson, J. D. G., Lueck, C. J., Friston, K. J., Kennard, C., & Frackowiak, R. S. J. (1991). A direct demonstration of functional specialization in human visual cortex. *The Journal of Neuroscience*, *11*(3), 641–649.